UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS FOR OCULAR SCANPATH PREDICTION

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

CAMILO ALEJANDRO JARA DO NASCIMENTO

PROFESOR GUÍA:
MARCOS ORCHARD CONCHA

PROFESOR CO-GUÍA:
CHRIST DEVIA MANRÍQUEZ

MIEMBROS DE LA COMISIÓN:
JORGE SILVA SÁNCHEZ
PEDRO MALDONADO ARBOGAST

Este trabajo ha sido parcialmente financiado por FONDECYT 1210031, Fundación Guillermo Puelma, la infraestructura de supercómputo del NLHPC (ECM-02) e Iniciativa Científica Milenio (ICM-P09-015F)

SANTIAGO DE CHILE
2022

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
Y AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: CAMILO ALEJANDRO JARA DO NASCIMENTO
FECHA: 2022
PROF. GUÍA: MARCOS ORCHARD CONCHA PROF. CO-GUÍA: CHRIST DEVIA MANRÍQUEZ

APLICACIONES DE REDES NEURONALES ARTIFICIALES PARA LA PREDICCIÓN DEL SCANPATH OCULAR

Presentamos un estudio de varios modelos neuronales que predicen los recorridos de escaneo oculares humanos (scanpaths) mientras visualizan libremente diferentes tipos de imágenes, y un análisis de qué arquitectura logra los mejores resultados. Esta comparación se realiza analizando diferentes métricas para compararar scanpaths, éstas tienen como objetivo medir errores espaciales y temporales; tales como MSE, ScanMatch, peaks de correlograma cruzado y MultiMatch. Nuestra metodología comienza eligiendo una arquitectura y entrenando diferentes modelos paramétricos por sujeto y tipo de imagen, esto permite que los modelos se ajusten a cada persona y conjunto de imágenes dado. Descubrimos que existe una clara diferencia en la predicción cuando las personas ven imágenes con alto contenido visual (contenido de alta frecuencia) y bajo contenido visual (contenido sin frecuencia). Las mejores características de entrada para predecir los scanpaths son los mapas de saliencia calculados a partir de imágenes foveadas junto con el scanpath ocular de los sujetos, esto modelado por nuestro modelo FovSOS-FSD.

Los resultados de este estudio podrían usarse para mejorar el diseño de interfaces controladas por la visión, realidad virtual, comprender mejor cómo los humanos exploran visualmente su entorno y allanar el camino para futuras investigaciones.

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
Y AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: CAMILO ALEJANDRO JARA DO NASCIMENTO
FECHA: 2022
PROF. GUÍA: MARCOS ORCHARD CONCHA PROF. CO-GUÍA: CHRIST DEVIA MANRÍQUEZ

APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS FOR OCULAR SCANPATH PREDICTION

We present a study of several neural models that predict human ocular scanpaths while they are free-viewing different images types, and an analysis of which architecture achieves the best results. This comparison is made by analyzing different metrics that encompass scanpath patterns, these metrics aim to measure spatial and temporal errors; such as the MSE, ScanMatch, cross-correlogram peaks, and MultiMatch. Our methodology begins by choosing one architecture and training different parametric models per subject and image type, this allows to the models adjust to each person and given set of images. We find out that there is a clear difference in prediction when people free-view images with high visual content (high-frequency content) and low visual content (no-frequency content). The best input features for predicting the scanpath are saliency maps calculated from foveated images together with the ocular scanpath of subjects, modeled by our FovSOS-FSD model.

The results of this study could be used to improve the design of gaze-controlled interfaces, virtual reality, as well as to better understand how humans visually explore their surroundings and pave a way to make future research.

Agradecimientos

Y para terminar (ya que para transmitir mejor el mensaje de lo agradecido y feliz que estoy puede ser descrito solamente luego del fin de escrito). Agradecer a todos los que me ayudaron en esta étapa de mi vida, estuvieron en las buenas y malas. Siempre. Mi familia. Tanto de sangre como no consanguineo, aquellos que ya no están con nosotros y aquellos que han comenzado a formar parte de mi familia. Mis viejos, mi hermano, mi vovo, mi yaya, chikon litol, niki jom, pastoral buchef y brraxitos de electrica. Muchas gracias a todos los que me guiaron y enseñaron. Agradecer a mis profesores y colegas que me dieron las herramientas para avanzar en mi sueño, seguiré avanzando en aprender sobre nuestro cerebro... Muchas gracias por formarme!

Y para comenzar ¿Crees poder describir lo que piensas. Es posible. Puedo sentimentalizar esto como un oxímoron, el termino del comienzo. En mi caso, siempre queda algo, y esa sensación de nunca terminar el comienzo es la conflictiva en la mayoria de los casos... esa inconformidad que no nos permite acabar. Al menos creo que me he acercado un poco. Y la sed de continuar no se acabará, el mayor lamento que queda es el tiempo que no permitirá realizar todo, o no? ia k saleeeeeeeeeeeeeeeeeeeee!

GPT3: Pienso que es posible describir lo que se piensa, pero que esto dependerá de la capacidad de expresión de cada persona. Para mí, poder describir lo que se piensa es como una forma de conectar con las demás personas, de entenderlas y de ser entendida. Creo que esto es especialmente importante cuando se trata de temas complejos, ya que permite que el diálogo se construya sobre bases sólidas.

Contents

List of Figures

Chapter 1

Introduction

Recent studies theorize the brain is constantly updating a mental model of the environment by minimizing an error function [Friston et al., 2006, Friston, 2013, Clark, 2013, Carhart-Harris et al., 2014], this is done by feed-forward connections which are speculated to carry the residual errors between the predictions and the actual lower-level activities [Rao and Ballard, 1999]. If we could stimulate these forward connections at will by creating a situation of uncertainty, this could force the propagation of residual errors, and thus we would be closer to demonstrating the theory raised above. We believe that eye movements influence visual prediction by modulating the associated brain activity. For instance, we create ANN models for predicting people's scanpaths, so that in further studies we could modify the area that people will see to break their brain's expectation, and thus stimulate their feed-forward connections which carry the signal residual error.

1.1 Motivation

The brain can be seen as a predictor system that based on the input signals provided by its sensors, estimates the future according to its present information by minimizing the prediction error; this is known as the brain's predictive ability or Predictive Coding. [Hosoya et al., 2005] found the spatio-temporal receptive fields of retinal ganglion cells encode local differences in space (rather than the raw image intensity) and this receptive field change in time (after a few seconds in a new environment), this new receptive field improves predictive coding under the new image statistics, this can be seen as a strategy of predictive coding adapted through evolution to the average image statistics of the natural environment. In this regard, it is speculated that the brain generates expectations by approximating its posterior distribution in a similar way as the Bayes theorem does, in other words, the brain would operate as a variational Bayesian entity.

One way to formalize this approach is through the Free Energy Principle (FEP) [Friston et al., 2006], which explains how living systems are kept in energy balance despite the cost of staying alive. The free energy is a function of the sensory input and an approximate probabilistic representation (recognition density) over unobserved variables of the environmental state. In statistics, this function is known as the Evidence Lower BOund (ELBO) [Yang, 2017] and it can be expressed as the negative energy plus the entropy. Then, the FEP minimization[1] induces the recognition density

[1]This is the same as maximizing ELBO due to ELBO is the additive inverse of the free energy.

to resemble the true posterior probability (as it desired when approximate through variational Bayesian inference), this is precisely what arises from the experimental evidence in Neuroscience.

We aim to find a Bayesian approach that can model brain function, therefore, it is of interest to investigate the modification of the nervous system when it is in situations where the uncertainty of the future changes abruptly for the observer. Our work seeks to pave this path, trying to force these kinds of situations that generate high uncertainty in the brain. We hypothesize that eye movements influence visual prediction by modulating the associated brain activity, for instance, if we find a model that predicts where a subject will observe, we will be able to modify that area beforehand and thus disturb the subject's expectation, forcing this situation of cerebral uncertainty.

The model generated in this research will allow future research (not part of this thesis) to perform behavioral and electrophysiological experiments to demonstrate that the brain works as a Bayesian entity that tries to minimize free energy.

1.2 Hypotheses

Our work is based on the following hypotheses:

- The selection of where a subject will look is modulated by where the eye movements were located. For instance, where a subject will look is influenced by where this subject saw.
- Salient objects in scenes modulate where eye movements will pay attention. The salient objects are chosen by what a specific subject considers most important to attend, conditioned by the individual subject's past experiences.
- The fovea influences how a subject considers an object to be more salient and relevant. This is by perceiving the details of an image (visual acuity) and allowing the subject to focus.

1.3 Objectives

1.3.1 Main Objectives

The main objectives of our work are the following:

- To implement an efficient and precise neural network model to predict eye movements.
- To design, implement and select the most relevant information to feed our neural network model to predict eye movements.

1.3.2 Specific Objectives

The specific objectives of our work are the following:

- To determine the best features to incorporate as inputs in predictive models.
- To determine if the knowledge of the ocular scanpaths is sufficient information for prediction purposes.
- To incorporate knowledge on the saliency in a scene and the subject ocular scanpath to predictive models. The saliency calculation should be dependent on where the subject fovea

is located.

- To characterize the prediction error in models trained with scanpaths that are obtained from the free exploration of different scene types.

1.4 Thesis Structure

The structure of our work is as follows:

- Chapter 2 presents a brief review of the ocular scanpaths. Describing the different eye movements types. Then, how the people's ocular scanpath is affected by internal and external factors conditioning where subjects will look. Finally, we define the metrics we are going to use to measure the differences between ground truth and the predicted scanpaths.
- Chapter 3 presents the most relevant works that addressed the scanpath prediction. Then, describes the theoretical background for understanding how neural networks work and how we can use them using different forecast strategies.
- Chapter 4 is the main core of our work where we address the problem of scanpath prediction by creating many artificial neural models. We present three main models and measure their performances using the metrics defined before.
- Chapter 5 presents the main conclusions of our developed work, provides some guidelines for future perspectives and improvements that could be made to our proposed models.

Chapter 2

Scanpath Ocular

Humans collect visual information using their eyes looking at the environment, due to anatomical limitations we can gather high-level information or visual details only in a small central area of our field of vision called the fovea. A central aspect of studies in this area includes two types of eye movements; rapid eye movement (saccades) and maintaining focus on a region (fixations). Even when fixating, the eyes are never completely at rest, they make frequent micro-saccadic eye movements in the fixated zone. Several factors can influence eye movement when viewing a scene: 1) the task being performed by the viewer and their intrinsic knowledge (top-down factors) and 2) the properties of the image being viewed (bottom-up factors). Typically, when presented a scene viewers demonstrate short fixation durations and long saccadic amplitudes in the early stages of viewing an image, followed by longer fixations and shorter saccades in the later stages of scene viewing processing [Pannasch et al., 2008], it also has been found that fixation durations and the length of saccades change as age increases [Helo et al., 2014]. In addition, the horizontal and vertical eye movements are found to behave differently [Rottach et al., 1997].

2.1 Eye Movements

Eye movements are used to gather more information from the surrounding environment. The eyes can scan a scene by moving rapidly from one point to another, or they can fixate on a single point and move only the eye muscles to change the angle of the lens. These movements can be classified according to three categories:

1. The involvement of one or both eyes:
    - Duction: one eye moving.
    - Version: both eyes moving in same direction.
    - Vergence: both eyes moving in opposite direction.
2. Gaze-stabilizing
    - Fixations.
3. Gaze-shifting mechanisms:
    - Saccades.
    - Pursuit movements.

For our work scopes, we focus on two types of movements for representing the subjects' scanpaths. First, and the most common type of eye movement is the saccade; a saccade is a fast, jerky eye movement that is used to change the direction of gaze. It occurs when a person looks at an object that is not in the center of their visual field. The second type of eye movement we focus on, are the fixations; a fixation is a span when the eyes are held still (whilst making micro-saccades around the area where the eyes stare).

2.2 Brain Selection

A common thought about how the human brain selects where to look is that it first generates a full scanpath and then executes fixation by fixation. However, neuroscience results suggest that as the eye fixates to a location on an image, the brain selects the next point to be seen and then executes the next saccade, after which the process is repeated [Kalesnykas and Sparks, 1996, Girard and Berthoz, 2005, Krauzlis et al., 2013]. It is speculated that our brain samples a sub-optimal Bayesian path to look at by minimizing the energy cost of choosing it [Najemnik and Geisler, 2009, Najemnik and Geisler, 2008], in other words, saccadic plans were influenced by the attempts of minimizing the cognitive and attentional load [Araujo et al., 2001].

2.3 Spatial Selection

The eye movements are affected by both bottom-up and top-down factors, even an initial glimpse of a scene influences subsequent eye movements [Castelhano and Henderson, 2007]. The bottom-up factors like the local contrast, the saliency or conspicuity of objects in the visual environment affect where the fixations will be [Itti and Koch, 2000]. Furthermore, a large contrast in luminance [Parkhurst et al., 2002] or a greater density of edges [Mannan et al., 1996] can affect the guidance of eye movements, also it has been found that the variance in fixations in natural scenes can be explained by local scene color [Amano and Foster, 2014]. On the other hand, researchers found that top-down factors (features) on scenes have a greater impact on predicting where eyes will fixate, interesting locations or saliency was a better feature than any low-level image feature (bottom-up factors) and any pair-wise combination between them [Onat et al., 2014].

[Henderson et al., 1999] found that the initial fixation placement is not controlled by the analysis of individual objects in the scene and once an object has been fixated the eyes tend to fixated longer on objects that are semantically informative than uninformative in the context of the scene, also Henderson and colleagues found that eyes tend to return to objects that are semantically inconsistent with the scene. Eye movements can also be guided towards items when they are heard verbally at the same time as seeing them i.e. the concurrent linguistic input may influence the decision of where to move the eyes [Staub et al., 2012]. Additionally, cultural differences in perceptual judgment and memory, it has been found that Westerners have an inclination to concentrate on focal objects in a scene, whereas East Asians attend more to contextual information [Chua et al., 2005].

2.4 Fixation Duration

Average fixation durations last for about 300 ms approximately, although there is a good deal of variability around this mean [Henderson, 2003, Rayner, 1998, Henderson and Hollingworth, 1998, Salthouse and Ellis, 1980]. This variability is mostly due to the properties of an image and in the task being done, which impact both bottom-up and top-down processing. Much of this variability is controlled by visual and cognitive factors associated with the currently fixated scene region, i.e. the individual fixation duration is affected by scene luminance [Loftus, 1985] and contrast [Loftus et al., 1992]. Along with the above, individual fixation durations in a scene were longer when the image at fixation was reduced by contrast or partially obscured by a noise mask, suggesting that fixation duration is influenced by the acquisition of visual information from the currently fixated region [Van Diepen et al., 1995, Van Diepen et al., 1998]. Individual fixation durations are also influenced by viewing task, with longer fixation durations during scene memorization than search [Henderson et al., 1999]. First-pass gaze durations are also influenced by object and scene semantics, with longer gaze durations on semantically informative (less consistent) than uninformative (more consistent) objects [Henderson et al., 1999].

2.5 Saliency and Selective Order

During the fixation period, humans extract the visual information necessary to perform an analysis of the scene. Given this, not all zones will be taken into account with the same importance, this is known as **saliency** or **visual attention**, so saliency reflects what people consider important and relevant in a scene. On the other hand, not all areas of a scene will be seen in the same order when comparing between subjects, this feature is known as **selective order**, based on the past where people will look.

Saliency defined by [Koch and Ullman, 1985] as a representation of conspicuity for every location in an image, has been studied for decades by studying the spatial density of fixations and salient objects in a scene [Koch and Ullman, 1985, Bruce et al., 2016, Islam et al., 2017, Islam et al., 2018]. For scanpath prediction models, saliency has been widely used in the literature when the problem is considered as a static distribution from which a sample is taken to obtain sequential fixations. The saliency prediction algorithms have improved significantly over the years, however, the vast majority of tests have been done with datasets that have few salient objects, on the other hand, by increasing the number of objects the attentional behavior of people vary significantly from one to another, for instance, sequential analysis emerges as the best solution to this problem [Fahimi and Bruce, 2021].

Selective order has not been sufficiently exploited in the literature, but attention to objects has become increasingly relevant in the context of deep learning in which recurring mechanisms drive the gradual solution of this problem [Xu et al., 2015] and give us an idea of how to add this selective order on our models.

2.6 Calculating saliency

Saliency models predict the probability distribution of the next locations that the eye will fixate on a scene, this distribution is known as the saliency map. Saliency has attracted a lot of interest as

it can provide insights into how the human attends and apply it to machine learning applications.

The basis for the creation of saliency models was laid on the theory of feature integration [Treisman and Gelade, 1980]. This theory was exploited by [Koch and Ullman, 1985] and [Itti et al., 1998] to start with a new way of predicting saliency given any arbitrary image without the need of pre-computing elementary features since then, numerous computational models have been proposed.

The above paved a way for many saliency prediction algorithms and models: [Harel et al., 2006] proposed to extract multi-scale low-level features from the scene to predict saliency based on graph algorithms. [Bruce and Tsotsos, 2005] predicts saliency based on the principle of maximizing information sampled from a scene. [Kienzle et al., 2007] addresses the bottom-up influence of local image information on human eye movements where the model directly learns from human eye movement data. [Zhang et al., 2008] adopt a Bayesian framework from which bottom-up saliency emerges naturally as the self-information of visual features and the overall saliency is the point-wise mutual information between the visual features and the desired target.

Recently, new models have been created based on Deep Neural Networks (DNNs), these networks allow to automatically learn an image representation achieving astonishing results solving tasks such as object detection, image segmentation, and image classification [Redmon and Farhadi, 2017, Simonyan and Zisserman, 2014, Girshick, 2015].

In this sense, DNNs helped improve saliency prediction [Xu et al., 2014] and several computational models have incorporated object detectors in their saliency models [Kümmerer et al., 2016, Huang et al., 2015, Cerf et al., 2008].

In general, the former prediction algorithms have in common that they calculate a saliency map using low-level features from a scene instead of learning from subjects' data what is the most salient region. Whilst the latter, DNNs learn a parametric representation of saliency maps from data. One advantage when using DNNs as saliency predictors is that we can fine-tune the network if we have data, so for example we can create our own saliency map predictor DNN network based on where subjects saw the most.

For our purposes, we think of a way to obtain different and fine representations of foveated images, we focus on the saliency maps obtained from models based on DNNs. This is because at the slightest change on the image, such as two images with foveations in different areas, DNNs find two saliency maps that are consistent on where the foveations were located.

In the next chapters, we will use the SALICON model for saliency estimation, this model was created by [Huang et al., 2015] based on DNNs architecture, see Figure 2.1 for details. This model got its apprenticeship by adjusting its weights initialized from a pre-trained DNN applied at two different image scales (pre-trained networks were trained on ImageNet [Deng et al., 2009]). We chose this model even though it was not the state-of-art in saliency map estimation like [Linardos et al., 2021, Kümmerer et al., 2016], because we found a direct implementation of SALICON in Tensorflow that fit our code, we prefer to leave as future work to test other saliency map estimators.

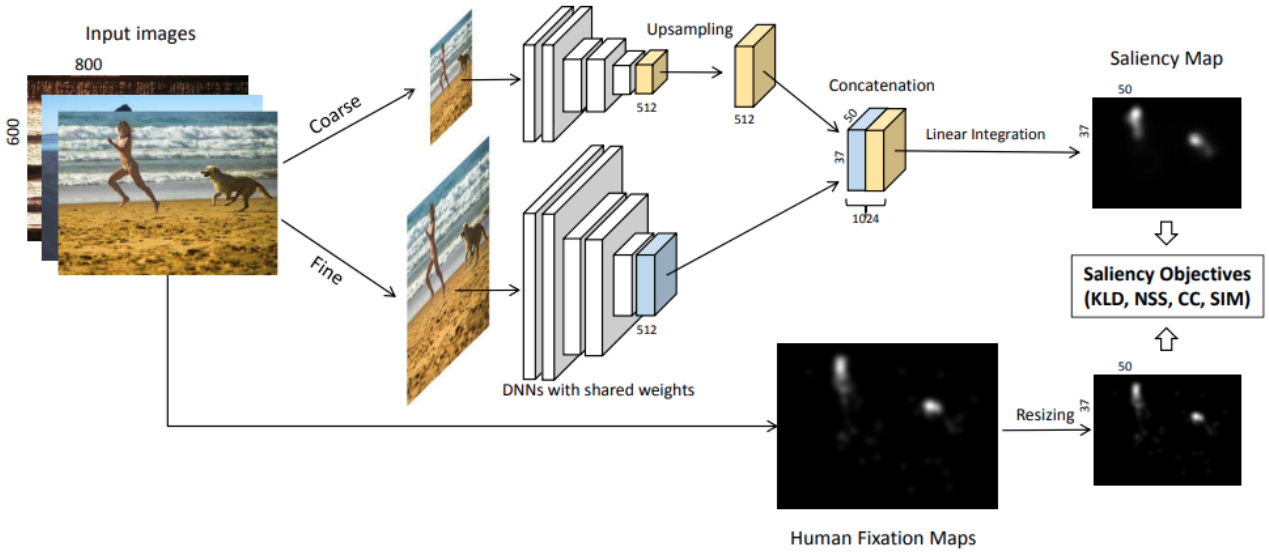

Figure 2.1: SALICON model used to retrieve saliency maps. SALICON learning procedure of the DNN architecture to estimate saliency. Diagram extracted from [Huang et al., 2015].

2.7 Fovea centralis

The fovea is a small central pit in the retina of the eye. It is responsible for sharpening the central vision and is the area of the retina that has the highest concentration of cone cells. This allows for this zone to have the highest visual acuity necessary in humans for activities that involve visual details.

As the highest visual acuity is just a small region of about 2.5 visual degrees, subjects cannot see the entire scene at the same resolution. Therefore, saccades and fixations are treated differently. When a saccade is performed it produces a shift in the retina that causes consequently that the fovea changes its focusing region. On the other hand, when fixating the fovea remains almost in the same zone observing high-resolution details.

Since we all know our attention is constantly shifting when free-viewing a scene. We think our brain is generating constantly saliency maps for searching tentative areas to look at, and this variability of saliency maps is given by the location of the fovea.

We believe that if our model uses the information of the fovea for generating saliency maps, it will help our model to better predict scanpaths. Therefore, the model could better understand where a person is likely to look next while viewing a scene. In this way, the model could take into account that when a subject is looking at a scene their attention is not equally distributed across the entire scene, but is instead focused on specific regions displayed on the saliency maps.

2.8 Scanpath Comparison Metrics

The problem of scanpath prediction has been mainly studied by considering the problem as a static distribution of rapid gaze movements and fixations. The vast majority of models have tried to learn the regions of the scene that the subjects usually see and, with this information, try to predict the entire scanpath sequence. However, our attention and saliency are not only dependent on the initial spatial distribution but what we consider most salient constantly changes with the observer gaze.

Even more, our brain is constantly considering the temporal dynamics of saliency while predicting where a subject is likely to look next. This selection of where to look is aided by our attentive mechanism which made this process sequentially and dependent of what it was seen before.

Given the above, most models and analyzes to date have not taken the temporal dynamics while predicting saliency, since they consider that there are equally interesting regions in a scene and no particular order to choose any region. Although some efforts have been made to analyze sequential fixations with sequential models, the existing metrics fail to adequately capture the performance of the model and certain metrics are revealed to be much more discriminatory than others [Fahimi and Bruce, 2021].

In this section we aim to review some useful metrics that have been used in our study to measure the difference between the real scanpath and the predicted one, these measures can be classified into five categories [Fahimi and Bruce, 2021]:

- Direct measures: based on Euclidean distance between positions of the ground truth and the predicted scanpath.
- Time-series analysis: based on comparing the displacement in time between two time-series, the ground truth, and the predicted scanpath.
- String-based metrics: based on binning the scanpath on the spatial scene, then codifying it into strings and applied string-based metrics for comparing the ground truth and the predicted scanpath.
- Vector-based metrics: based on the alignment of the scanpath for applying vector comparisons between the ground truth and the predicted scanpath.
- Recurrence analysis: based on describing dynamic systems by capturing global and local temporal characteristics of a sequence.

As there are a lot of similarity metrics to evaluate the differences between the ground-truth and the predicted scanpath, in our analysis we focus on the Mean Squared Error (MSE), the Cross-correlogram Peak, ScanMatch [Cristino et al., 2010] and MultiMatch [Jarodzka et al., 2010].

Definition 2.1 (Mean Squared Error) *An estimator that measures the average of the squares of the errors. This metric measures the error in amplitude, see equation (2.1).*

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2, \tag{2.1}$$

where Y is vector of the observed values and its predicted values $\hat{Y}$.

Definition 2.2 (Cross-Correlogram) *The cross-correlogram a.k.a temporal correlation is an estimator that measures the auto-correlation of a time-series along time. This metric measure the displacement in time between the observed and the predicted values, see equations (2.2, 2.3)*

Consider the coefficient of auto-correlation at a lag h is given by $r_h = \frac{c_h}{c_0}$, where c_h is the auto-covariance and c_0 the variance between the observed values Y and the predictions $\hat{Y}$*:*

$$c_h = \frac{1}{N}\sum_{t=1}^{N}(Y_t - \hat{Y})(Y_{t+h} - \hat{Y}), \tag{2.2}$$

$$c_0 = \frac{1}{N}\sum_{t=1}^{N}(Y_t - \hat{Y})^2. \tag{2.3}$$

Definition 2.3 (ScanMatch) *ScanMatch was formulated by [Cristino et al., 2010] for comparing scanpaths, this metric solves some of the shortcomings of the edit distance (a.k.a Levenshtein distance) [Levenshtein, 1966] in considering semantic information and duration of fixations. ScanMatch follows similar pre-processing steps of the edit distance for converting scanpaths to a sequence of strings but in ScanMatch the fixation duration points are taken into account by quantizing it using a fixed temporal bin for repeating the character [Fahimi and Bruce, 2021]. Figure 2.2 show how the transformation process from scanpath to string sequence is done, then for compare and align string the Needleman-Wunsch algorithm is performed [Needleman and Wunsch, 1970].*

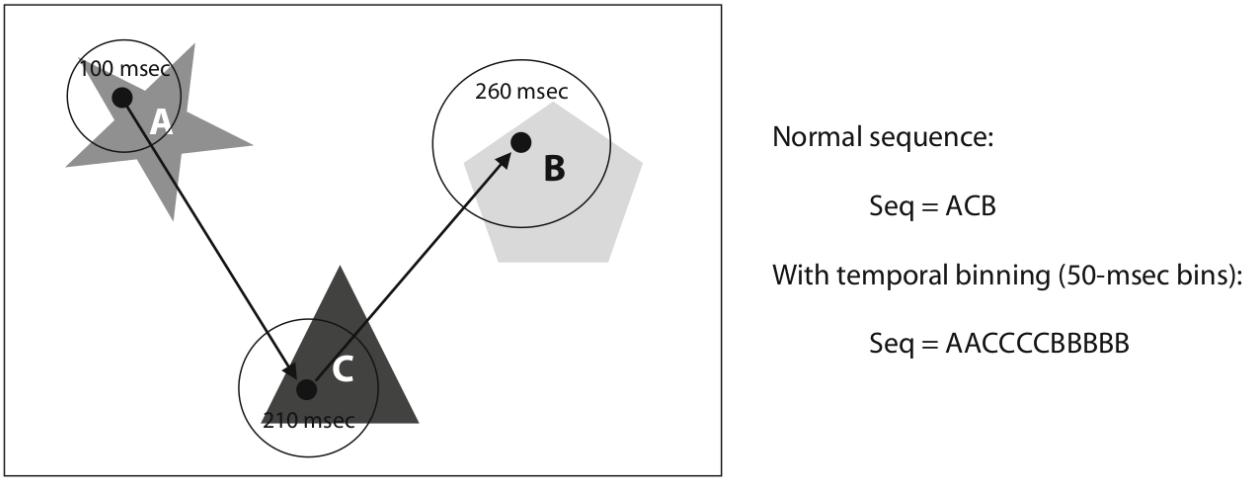

Figure 2.2: ScanMatch metric is used to measure spatial and temporal differences between scanpaths, by converting scanpaths to a sequence of strings and taking into account the fixation duration with temporal binning. Diagram extracted from [Cristino et al., 2010].

Definition 2.4 (Multimatch) *Multimatch defines scanpaths as a series of geometric vectors in space, scanpaths are temporally aligned and then compared across several dimensions [Jarodzka et al., 2010, Dewhurst et al., 2012]:*

- *Shape: Vector difference between aligned saccade pairs.*

- *Direction: Angular distance between aligned saccade pairs.*
- *Length: Difference in length between the endpoints of aligned saccade pairs.*
- *Fixation position: Difference in position (euclidean distance) between aligned fixations pairs.*
- *Fixation duration: Difference in duration between aligned fixations pairs.*

Our analyses will be focus on the above mentioned metrics for the following main reasons:

- The MSE is for measuring the spatial difference in error between the predicted and the ground truth scanpaths.
- ScanMatch is an improved version of the edit distance that compares the saccadic eye movement as sequences. Among its improvements, we find that it measures both temporal and spatial differences between the predicted and the ground truth scanpaths.
- The peaks of the cross-correlogram allow us to compare the temporal phase's gap, that is, measuring the time lag difference between the ground truth and the predicted scanpaths. This measure is very sensitive to temporal and spatial differences between the two scanpaths and can be advantageous because precise temporal timing of gaze sequences is crucial [Anderson et al., 2015].
- The Multimatch algorithm compares the predicted and the ground truth scanpaths on five different aspects: vector shape, vector length (saccadic amplitude), vector position, vector direction, and fixation duration. This allows us to have a complete battery of metrics for checking every aspect when we compare two scanpaths.

Chapter 3

Theoretical background for Scanpath Prediction

As it was shown in the previous chapter, how we choose where to look is a question that has attracted a lot of research over the decades. In past years, there has been an increase in models creation that aims to predict human ocular scanpaths when free viewing images.

This chapter briefly summarizes the work done in estimating and predicting scanpaths. First, we will make an overview of the models that reach the state of art in scanpath prediction. Then we will focus our attention on a particular area of research of these models, the engineered models that are based on Artificial Neural Networks (ANNs), for that we explain how artificial neural models works, and the advantages and disadvantages of their components. Finally, we show the ways to make a multi-step forward prediction for any time series that in our case will be the human ocular scanpath.

3.1 Models for Scanpath Prediction

In the current section, we summarize the existing literature models that predict human scanpaths. Considering eye movements as a manifestation of an attentional decision process, the researchers take into account several assumptions and their ideas to create models like biologically inspired models, statistically inspired models, cognitively inspired models, and engineered models [Kümmerer and Bethge, 2021].

About biologically inspired models, the Itti-Koch model predicts saliency [Itti et al., 1998] and can be used as a scanpath predictor with its fixation selection mechanism which uses the winner-takes-all-network method [Koch and Ullman, 1985]. Wang and colleagues use in their model the salience of the central environment, inhibition of return, fixation revision, an algorithmic implementation of information transmission of the neural network and the forgetting effect of short-term working memory [Wang et al., 2011]. Other models implement attention through reinforcing the area around the current fixation with a decay time period [Engbert et al., 2015, Schütt et al., 2017]. Adeli and colleagues' model applies a retina transformation to the input image which blurs it locally depending on the last fixation, then is computed a priority map into superior colliculus

space [Adeli et al., 2017]. Zanca and colleagues use a differential equation system which has as input the gradient of brightness, the gradient of optical flow, and an inhibition-of-return potential updated through time [Zanca et al., 2019].

On the other hand, statistically inspired models, make assumptions about the scanpath distribution trying to estimate how it will evolve in time. [Brockmann and Geisel, 2000] assume that next saccades to be distributed proportionally to the product of a saliency potential and a functional that depends on the distance of the last fixation, they find that this process is described by a Cauchy distribution. Following this discovery, [Boccignone and Ferraro, 2004] models saccades lengths with a Cauchy distribution but the ballistic jump (jump distribution) of saccades was modeled with saliency maps. [Sun et al., 2014] opt to use a Gaussian component analysis over patches from the images to select the fixation with its respective highest response of the Gaussian component. [Clarke et al., 2017] implements a Gaussian jump distribution that fits its parameters based on the previous fixation distribution. [Coutrot et al., 2018] model the scanpaths using images and a Hidden Markov Model (HMM) with 2-dimensional Gaussian states. [Xia et al., 2019] propose an iterative representation learning model (IRL) that uses the saliency of images locations obtained from an auto-encoder artificial neural network. At last, we want to notice that these kinds of models have the disadvantage of assuming a distribution of how the scanpath will evolve, which in general might not be well adjusted to a specific person but only to a population.

There are also many models based on cognitive aspects, it is known that eye movements are attracted by low-level features like edges and contrast and by high-level content such as objects and faces [Kummerer et al., 2017]. Hereby, Liu and colleagues based on three principal factors that influence human attention create their model with low-level saliency features, spatial positions, and semantic content [Liu et al., 2013].

Then, due to the recent advances in Deep Learning, researchers try to incorporate this approach into their models, allowing them to learn from observations and adjust to specific subjects. Assens Reina and colleagues create SaltiNet to predict spatiotemporal saliency by using as a prior a jump distribution for saccadic prediction [Assens Reina et al., 2017]. Two years later, Assens and colleagues propose its PathGAN model based on a generative adversarial network architecture with a recurrent neural network with spatiotemporal sequences as input [Assens et al., 2018]. DeepGaze II [Kümmerer et al., 2016] uses transfer-learning with features extracted from the VGG19 [Simonyan and Zisserman, 2014] network to compute saliency maps that encodes the last two fixation locations.

In conclusion, scanpath estimation is a complex task that has been tackled by many researchers in the past years. Different methods have been proposed, each with its advantages and disadvantages. The use of artificial neural networks has shown promising results and is likely to play a bigger role in the future of scanpath prediction research as it pave a way for learning from general and specific data, data that day by day increase their volume allowing to create better scanpath predictor models.

3.2 Summary of Artificial Neural Models

An artificial neural network (ANN) is a parametric function inspired on biological neurons that receive one or more inputs signals which are passed forward through the network model, then a

weighted sum is applied for each input followed by a non-linear activation function which allows discerning between two or more states, in other words, it splits the output space into as many classes as desired to let us make decisions. A biological analogy is the dendrites that act as the input vector in the ANN, where the former is responsible of select the values of the weights for every input by increasing or decreasing the ratio of the synaptic neurotransmitters, then the Soma acts as the summation over the weighted inputs and the Axon as the activation function that once it reaches certain potential the signal is transmitted. In the artificial neural model case, the parameters or weights are learned by an optimization process known as Stochastic Gradient Descent (SGD) which leverages the automatic differentiation algorithm backpropagation for calculating the gradients of the weights, such optimization tries to minimize a pre-defined error or loss function between the expected values and the predictions.

In this section, we aim to tackle these concepts and review some of the most used configurations for neural models.

3.2.1 Theoretical definition

Let $\mathcal{Z}$ be the domain of observations or inputs $\boldsymbol{z} \in \mathbb{R}^n$, $n \in \mathbb{N}$ and let $\boldsymbol{y}$ be the outputs, logits or labels that takes their values from it image set $\mathcal{Y}$. Note that for a regression task we need to impose that $\mathcal{Y} \subseteq \mathbb{R}^m$, $m \in \mathbb{N}$, on the other hand, for a classification problem we need that $\mathcal{Y}$ must be discrete i.e. a finite subset $\mathcal{Y} \subseteq \mathbb{N}$. The pair $(\boldsymbol{z}, \boldsymbol{y})$ are suppose to be independent and identically distributed (i.i.d.) which characterize a joint probability distribution $\mathbb{P}_{(\boldsymbol{Z},\boldsymbol{Y})}$, where $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are the random vectors which represents the observations and the predictions, respectively.

To understand the core functionality of the ANNs, we focus here on its components, such as the neuron, weights, and activation functions. Also, we review how these models are trained by learning the distribution of weights which minimize some error, and then made predictions based on.

Definition 3.1 (Neuron or Perceptron) *[Rosenblatt, 1958] A function $f : \mathbb{R}^n \to \mathbb{R}$ composed of a weight $\boldsymbol{\omega} \in \mathbb{R}^n$, a bias $\boldsymbol{b} \in \mathbb{R}$ and a non-linear activation function $\sigma : \mathbb{R} \to \mathbb{R}$, see Figure 3.1:*

$$f(z) = \sigma(\boldsymbol{\omega}^T \cdot z + \boldsymbol{b}). \tag{3.1}$$

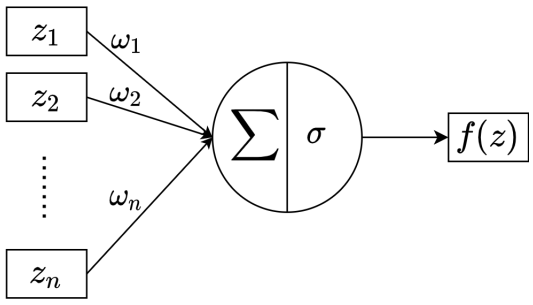

Figure 3.1: Perceptron is the core unit (neurons) of artificial neural network [Rosenblatt, 1958]. Perceptron diagram (own design).

Definition 3.2 (Dense Layer) *A set of m neurons* $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}^m$ *composed by weights* $\omega_i \in \mathbb{R}^n$*, biases* $b_i \in \mathbb{R}$ *and non-linear activations* $\sigma_i : \mathbb{R} \to \mathbb{R}$*, where* $i \in \{1, ..., m\}$*.*

$$\mathcal{L}(z) = \sigma(\boldsymbol{W}^T \cdot z + \boldsymbol{B}), \tag{3.2}$$

where $\boldsymbol{W} \in \mathbb{R}^{n \times m}$ *represents the weights matrix,* $\boldsymbol{B} \in \mathbb{R}^m$ *is the biases matrix and* σ *is a point-wise non-linear activation.*

For simplicity, we can also redefine the equation (3.2) by adding a one on the input and concatenate the biases matrix with the weights matrix:

$$\mathcal{L}(\tilde{z}) = \sigma(\widetilde{\boldsymbol{W}}^T \cdot \tilde{z}), \tag{3.3}$$

where $\widetilde{\boldsymbol{W}} = [W^T, B]^T \in \mathbb{R}^{n+1 \times m}$ *and* $\tilde{z} = [z, 1]^T \in \mathbb{R}^{n+1}$*.*

Definition 3.3 (Multi-Layer Network) *A function* F_ω *which depends on a set of parameters or weights* $\omega \in \Omega$ *that maps any sample* $\boldsymbol{z} \in \mathcal{Z}$ *to a point* $\boldsymbol{y}$ *of the output space* $\mathcal{Y}$*:*

$$\begin{aligned} F_\omega : \mathcal{Z} &\to \mathcal{Y} \\ \boldsymbol{z} &\mapsto F_\omega(\boldsymbol{z}), \end{aligned} \tag{3.4}$$

where F_ω is a composition of many neurons interconnected as a directed graph, this graph is organized in successive layers where the first layer is the input layer and the last layer are the outputs, therefore the middle layers are the hidden layers, see Figure 3.2.

$$\boldsymbol{y} = F_\omega(\tilde{z}) := \mathcal{L}^{(L)} \circ \mathcal{L}^{(L-1)} \circ \cdots \circ \mathcal{L}^{(1)}(\tilde{z}), \tag{3.5}$$

where $\mathcal{L}^{(l)}$ is the l-th layer of the network, $l \in \{1, ..., L\}$ and $\circ$ is the function composition.

3.2.2 Supervised Learning

Supervised learning is a type of machine learning algorithm where a model tries to learn how to perform a task from a set of training examples, and each example is paired with its respective target output. The goal of supervised learning is to learn a function that maps inputs (features) to the correct outputs (labels). For more details on the theoretical background explained in this section, refer to [Goodfellow et al., 2016].

3.2.2.1 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a powerful optimization technique that can be used to find the parameters of a machine learning model. The basic idea behind this technique is to iteratively improve the parameters or weights of a machine learning model by minimizing a cost function.

In each iteration, the algorithm samples a small number of data points from the training set and uses them to compute the gradient of the cost function with respect to the parameters of the machine learning model. This gradient is then used to update the parameters of the machine learning model.

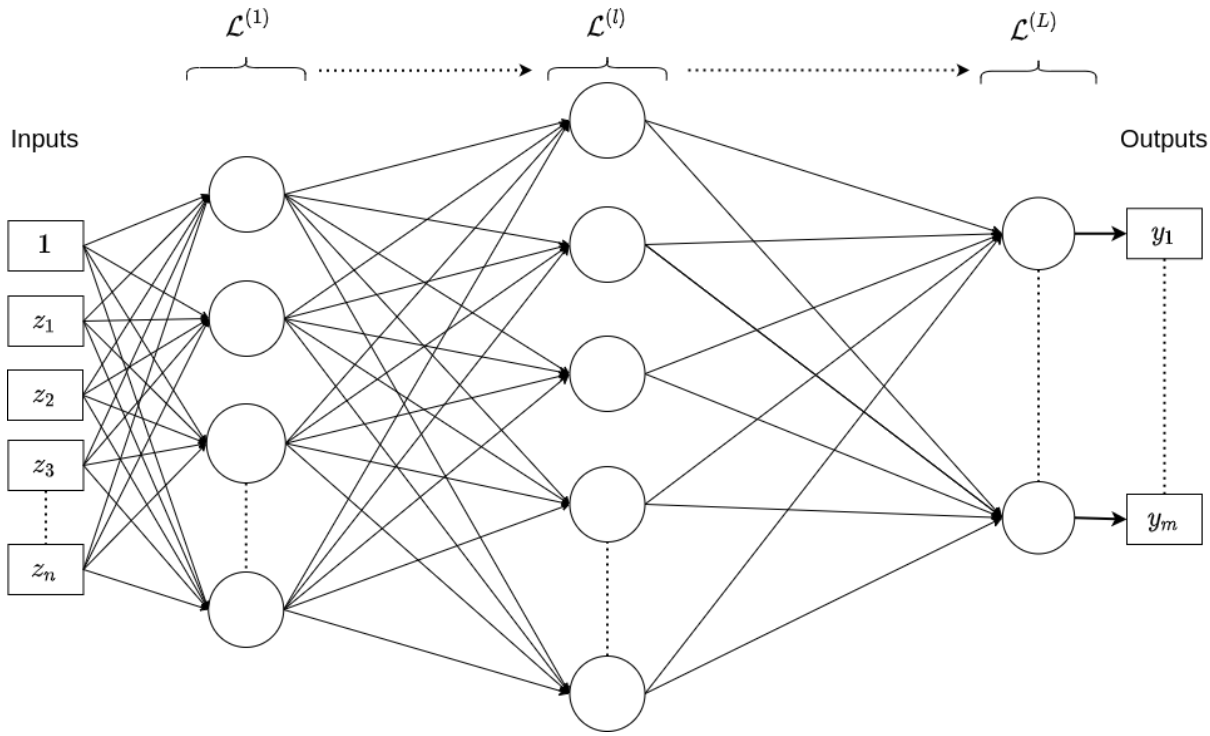

Figure 3.2: Multi-Layer Networks are composed of many neurons (perceptrons) grouped in consecutively layers. Multi-Layer Network diagram (own design).

The cost function used by a machine learning algorithm often decomposes as a sum over training examples of some per-example loss function:

$$J(\omega) = \mathbb{E}_{Z,y\sim p_{data}} L(z, y, \omega) = \frac{1}{N}\sum_{i=1}^{N} L(z_i, y_i, \omega), \tag{3.6}$$

where L is the per-example loss or cost function (for our work we use the MSE loss).

For these additive cost function, gradient descent requires computing

$$\nabla_\omega J(\omega) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\omega L(z_i, y_i, \omega), \tag{3.7}$$

As the computational cost is $\mathcal{O}(N)$ the time to take a single gradient step becomes prohibitively long as the training set size grows. For this, on each step of the algorithm, we sample a minibatch from the training dataset of size N'. Then, we can compute the gradient as follows:

$$g = \frac{1}{N'}\sum_{i=1}^{N'} \nabla_\omega L(z_i, y_i, \omega). \tag{3.8}$$

Finally, the stochastic gradient descent algorithm of the parameters have the following updating

rule:

$$\omega \leftarrow \omega - \varepsilon g, \tag{3.9}$$

where ε is the learning rate.

3.2.2.2 Backpropagration

Backpropagation is an algorithm most commonly used in machine learning for computing the gradients in a feedforward neural network. This algorithm propagates the gradients (of the cost function) back through the network by using a recursive chain rule, this allows us to use SGD in order to adjust the weights along with the network. By using backpropagation algorithm together with SGD it is possible to find sub-optimal[1] weights for our network.

The backpropagation algorithm is a recursive algorithm that can be broken down into the following steps:

1. Compute the cost function for the current input vector.
2. Compute the gradient of the cost function with respect to the weights in the network.
3. Propagate the gradient backward through the network, using the chain rule.
4. Adjust the weights in the network via SGD, so that the cost function is minimized.

For more specific details as the exact form of the recursive chain rule applied on neural networks, refer to [Goodfellow et al., 2016].

3.2.3 Neural networks

Neural networks are a type of machine learning algorithm that is inspired by the workings of the brain. They are composed of a large number of interconnected processing nodes, or neurons, that can learn to recognize patterns of input data. Neural networks can be used to solve a wide variety of tasks, such as image recognition, natural language processing, and machine translation.

For training neural networks it is necessary to feed the network with a training set of data, and then adjust the strengths (weights) of the connections between the neurons in the network in order to optimize a loss function. Neural networks can be trained using a variety of different methods, including gradient descent, and for calculating the gradients the backpropagation algorithm is used.

In this section, we briefly review some of the state-of-art in neural networks with their respective core layers, definition, and applications.

3.2.3.1 Convolutional neural network

The convolutional neural network (CNN) was created for analyzing images and their surroundings, this network was introduced by Yann LeCun in 1990 [LeCun et al., 1990] to solve image recognition tasks and they were inspired by the biological receptive field which responds to a stimulus only in a restricted region of the visual field. The CNN is composed of many convolutional layers where each

[1]Note that SGD gradients updates do not guarantee that the weights must be optimal, because they could reach a local minimum (not the global) as we use minibatches in the updating process.

neuron receives input from some number of locations that come from the previous layer (receptive field). Unlike Dense layers, the convolutional layer applies different filters or kernels on an input image which represents the weights that the network must learn.

Definition 3.4 (Discrete Convolution) *The discrete convolution defined on a set $\mathcal{Z}$ of integers, given f and g two complex-valued function is, see Figure 3.3:*

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f(m)g(n-m) \tag{3.10}$$

Definition 3.5 (Convolutional layer) *A parametric function $\mathcal{C}_K : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{\widetilde{H}\times\widetilde{W}\times\widetilde{\mathbb{C}}}$ which maps a 3-dimensional array (an image $I \in \mathbb{R}^{H\times W\times C}$) to another 3-dimensional array by applying $\widetilde{C}$ convolution filters ($K_{\tilde{c}} \in \mathbb{R}^{H_k\times W_k}, \tilde{c} \in \{1, ..., \widetilde{C}\}$) on every receptive fields of the image I. The above is done by moving the kernel along the image with certain steps known as strides:*

$$\mathcal{C}_K(i, j, \tilde{c}) = (K_{\tilde{c}} * I)(i, j) = \sum_{m=1}^{H}\sum_{n=1}^{W} K_{\tilde{c}}(m, n)I(i-m, j-n).^2 \tag{3.11}$$

From the definition 3.5, sometimes is desired to maintain the output size the same as the input, for it is possible to apply padding to the output image so we do not lose the image edges. Then, the spatial output size $(\widetilde{H}, \widetilde{W})$ depends on the following hyper-parameters of the convolutional layer: spatial input size of the image (H, W), kernels size (H_k, W_k), padding (P_H, P_W), stride (S_H, S_W)[3], for more details check the equation (3.12).

$$\begin{aligned} \widetilde{H} &= \frac{H + 2\cdot P_H \cdot (H_k - 1) - 1}{S_H} + 1 \\ \widetilde{W} &= \frac{W + 2\cdot P_W \cdot (W_k - 1) - 1}{S_W} + 1 \end{aligned} \tag{3.12}$$

3.2.3.2 Recurrent neural network

Recurrent Neural Networks (RNNs) are designed to process sequential data which is highly correlated through time [Rumelhart et al., 1986]. For instance, these kinds of networks have a memory about the past evaluations which were made, this is encapsulated by the hidden states h_t of the RNN. For the above, the architecture of an RNN is a cyclical graph that has a feedback loop that allows the temporal data $x_1, ..., x_\ell$ be processed.

[2]Note that equation (3.11) is very similar to (3.3), just replace the weights $\widetilde{W}$ by the kernel K and remove the activation function σ.

[3]Note that the lower index H and W denotes in which spatial dimension the operations (padding and stride) were applied.

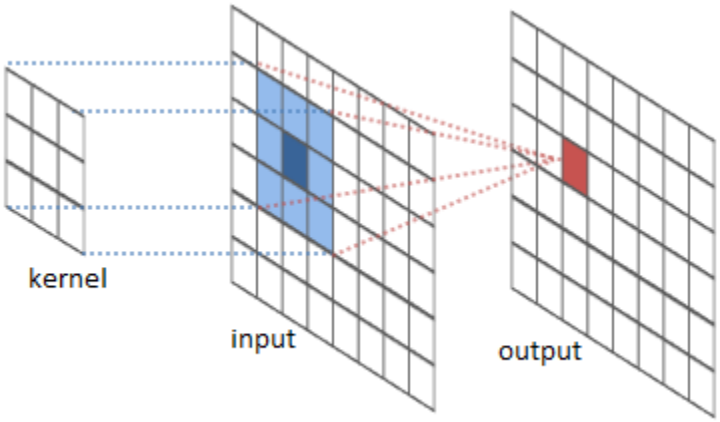

Figure 3.3: Convolution is the main operation used in convolution layers for processing images. Convolution diagram obtained from `http://intellabs.github.io/RiverTrail/tutorial/`.

Definition 3.6 (Hidden states) *A parametric function $\mathcal{H}_\omega : \mathbb{R}^{\ell \times H} \to \mathbb{R}^{\ell \times \tilde{H}}$ which maps a sequence of length ℓ with H features to another sequence of the same length as the input, but with $\tilde{H}$ features.*

$$h^{(t)} = \mathcal{H}_\omega([h^{(t-1)}, x_t]) \tag{3.13}$$

Definition 3.7 (Vanilla RNN layer) *A parametric model that learn and uses hidden states to make predictions $\hat{y}$ of sequential data.*

$$h^{(t)} = \sigma(b + W^T h^{(t-1)} + U^T x_t) \tag{3.14}$$

$$\hat{y}^{(t)} = softmax(c + V^T h^{(t)}) \tag{3.15}$$

where the parameters are the bias vectors b and c along with the weight matrices U, V and W.

Definition 3.8 (Long Short-Term Memory or LSTM) *[Hochreiter and Schmidhuber, 1997] A parametric layer which is composed by gates that controls the information flux, these are known as forget gate, input gate, output gate and memory cell, mathematically refers to equation (3.16).*

$$\begin{pmatrix} f_t \\ i_t \\ o_t \\ \bar{C}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \tag{3.16}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \bar{C}_{t-1} \tag{3.17}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{3.18}$$

where the initial values C_0 and h_0 are uniformly initialized, σ is the sigmoid activation function, $\otimes$ is the point-wise multiplication operator and W is the weights matrix containing the sub-matrices W_i, W_f, W_o and $W_{\bar{C}}$.

About the LSTM defined in 3.8, the forget gate f is in charge of deciding what information is important to keep or forget, whilst the input gate i decides the values which will be updated and the memory cell creates a feature $\hat{C}$ to aggregate them into the next state h, then the output gate o choose which values are selected for taking a decision.

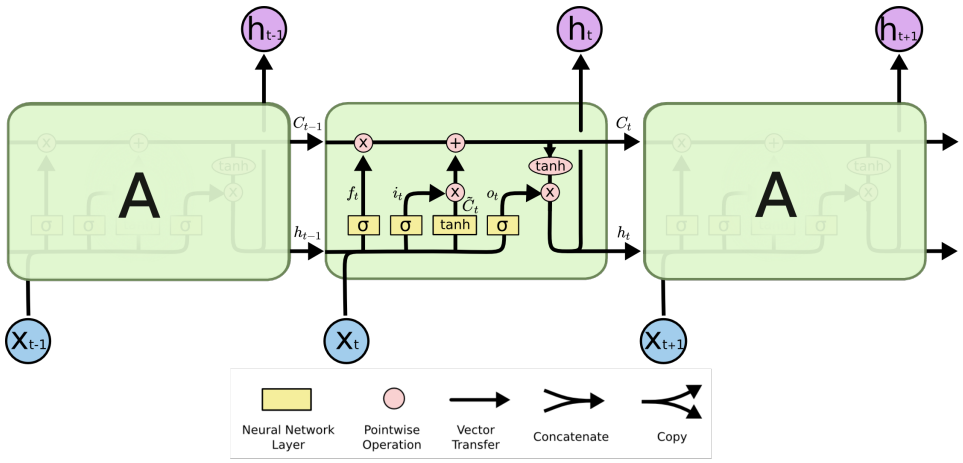

Figure 3.4: LSTM is designed to process sequential data through time. LSTM diagram obtained from `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

3.2.3.3 Attention neural network

An attention neural network is a parametric function that computes the compatibility between its inputs, this function mimics our cognitive attention which tries to enhance the important part of the input data and fades out the rest. The attention techniques used for the above are the *dot-product attention* which uses the dot product between vectors to determine similarity and the *multi-head attention* which combines the attentions or "heads" that were calculated in the dot-product attention stage.

Definition 3.9 (Scaled Dot-Product Attention) *[Vaswani et al., 2017] Its a function that takes queries Q and keys K of dimension d_k and values V of dimension d_v, and computes the compatibility of the query with the corresponding key*

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}V\right) \tag{3.19}$$

The Scaled Dot-Product Attention calculates the dot product between the query and key, this result is scaled by $\sqrt{d_k}$. With the previous step, we obtain the well-known attention vectors that determine how similar are queries Q and keys K by implicitly calculating the cosine between them (which ranges between -1 to 1, indicating low similarity and high similarity respectively). Next, every attention vector serves as weights that help us to know in what percentage the values V need to be attended, this process is made by multiplying the values V with its respective attention vector and then sum the results obtaining the output vector. Lastly, a softmax function is applied on the output vector in order to get the attention probabilities, see Figure 3.5.

Definition 3.10 (Multi-Head Attention Layer) *[Vaswani et al., 2017] Is a parametric function which use h-times the scaled dot-product attention to learn its weights:*

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h) \cdot W^O \tag{3.20}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$ and W_i^Q, W_i^K, W_i^V are the trainable weights matrices.

The main idea of the Multi-Head Attention Layer is performing attention in parallel, so obtain multiple different outputs that attend to diverse aspects of its inputs, then concatenate its results to obtain a new attention feature vector, see Figure 3.5.

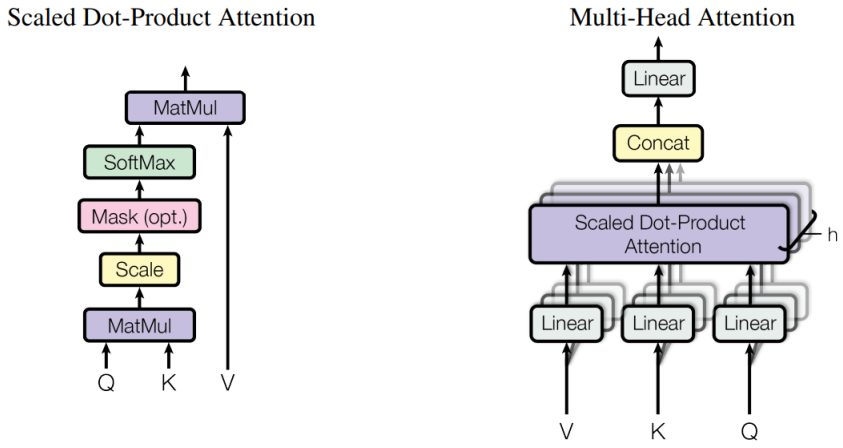

Figure 3.5: The main operations of Attention layers: the Scaled Dot-Product Attention and the Multi-Head Attention. Diagram obtained from [Vaswani et al., 2017].

3.2.4 Regularization methods

Regularization techniques are used in machine learning to prevent overfitting. One of the main scopes in machine learning is to fit models that generalize well to new data by reducing the test error.

3.2.4.1 Dropout

Dropout [Srivastava et al., 2014] is a technique to prevent overfitting in deep neural networks. It is achieved by randomly dropping some of the neurons in the network during the training process. This helps to prevent the network from learning the exact mapping between the input and output data and instead encourages the network to learn a more general representation.

The basic idea behind dropout is that if some of the neurons in the network are randomly dropped during the training process, the network will be forced to learn a more general representation. This is because the network will not be able to rely on specific neurons to learn the mapping between the input and output data.

3.2.4.2 Early stop

When training models, we commonly see that the training error decreases over time, this usually is the normal case. However, we always have to check if our validation set error increases because this will imply that the model is overfitting the data. The early stop is a strategy to avoid this case and stop the training phase if the validation set error begins to increase repeatedly over time.

3.3 Auto-regressive models for multi-step-ahead forecast in time series

In recent years, artificial neural networks have had many applications and solutions to problems in numerous areas of research. One of those areas is time-series forecasting which is the process of predicting future values of a time series. This process can be done in many ways, like using external data to make future predictions (the data used is extrinsic to the time series to predict), or using an auto-regressive approach that uses a combination of the past values from the same time series in order to make the future predictions. Another important aspect, is how much time ahead the prediction wants to be made, which is called the forecasting horizon. For our purposes, as we want to predict scanpaths we must adopt a multi-step-ahead approach to forecast the next h steps in the scanpath.

The multi-step-ahead forecast can be made in two ways. One of those is by estimating recursively the time series, that is computing the next prediction using the previous data and then using the predictions as "new available data" to continue the forecast until the desired horizon, this is called recursive forecast. The other way is that we can fit the model for predicting exclusively the desired horizon, this is called direct forecast.

Given a time series with ℓ data points $z_{t-1}^{(\ell)} = [z_{t-1}, ..., z_{t-1-\ell}]$ and the next observations set $\hat{Y}_h = \{z_t, ..., z_{t+h}\}$, where h is the horizon of the forecast, we want to optimize a parametric function f with a forecast error which has zero mean ε_t and a non negative variance σ^2.

$$\hat{Y}_h = f(z_{t-1}^{(\ell)}) + \varepsilon_t, \tag{3.21}$$

3.3.1 Recursive forecast

As we want to predict the next observations, the recursive approach predicts the next observation, then it is fed again in the parametric function until we predict the h-th next observation, we call it h-step-ahead (with a horizon h).

For the above, the parametric function with weights ω is optimized by stochastic gradient descent minimizing any loss (in our work we use the Mean Squared Error). In other words, we find the parameters ω^* which minimize the error between the targets and the predictions obtained by recursively feeding the parametric function.

3.3.1.1 Training phase

Let the training set $\mathcal{D}_{train}$ divided in batches $Z_{t-1}^{(\ell)} = \{z_\ell^{(\ell)}, z_{\ell+1}^{(\ell)}, ..., z_{t-1}^{(\ell)}\}$ where $z_t \in \mathbb{R}^n$ be the observation in the time t and the upper index indicates the length of the sequence, then $z_t^{(\ell)} = [z_t, z_{t-1}, ..., z_{t-\ell}]$ is a sequence of the previous ℓ observations. Let the respective targets of the training set defined by $Y_t = \{z_{\ell+1}, z_{\ell+2}, ..., z_t\}$, note that they are the next position of every sequence from the training set. For instance, we feed the network with every element of $Z_{t-1}^{(\ell)}$ and optimize the weights by finding the minimum MSE between the predicted positions $\hat{Y}_t$ and its respective targets Y_t,

$$\hat{Y}_t = f_\omega(Z_{t-1}^{(\ell)}) + \varepsilon_t, \tag{3.22}$$

$$\omega^* = \arg\min_\omega \sum_{\left(z_*^{(\ell)}, y \in Y_t\right) \in \mathcal{D}_{train}} (Y_t - \hat{Y}_t)^2, \tag{3.23}$$

3.3.1.2 Inference

To make recursive predictions with a horizon h, for every timestep t we use the previous ℓ observations $(z_t^{(\ell)})$ and feed them inside the parametric function for predict the next observation $\hat{z}_{t+1}$. Next, we take $\hat{z}_{t+1}^{(\ell)} = [\hat{z}_{t+1}, z_t, ..., z_{t-\ell}]$ for predict $\hat{z}_{t+2}$ and so on until reach h-steps-ahead i.e. predict $\hat{z}_{t+h}$. Note that there are 3 different cases dependent on the selected h and the sequence length ℓ: the first when $h = 1$ it just do a pass through the network and predict the next position, the second case it's when the horizon is smaller than the sequence length, the input it's a combination of data points containing the recursive predictions and the real data that is still available in the sequence, on the other hand, the third case are feed only with predicted data cause the horizon used is larger than the data available per sequence, see equation (3.24) for the general formula to recursively predict and in Figure 3.6 there is an application example.

$$\hat{z}_{t+h} = f_{w^*}^{[h]}(z_t^{(\ell)}) = \begin{cases} f_{w^*}(z_t^{(\ell)}), & \text{if } h = 1. \\ f_{w^*}([f_{w^*}^{[h-1]}(z_t^{(\ell)}), ..., f_{w^*}^{[1]}(z_t^{(\ell)}), z_t, ..., z_{t-\ell+h}]), & \text{if } 1 < h < \ell. \\ f_{w^*}([f_{w^*}^{[h-1]}(z_t^{(\ell)}), ..., f_{w^*}^{[h-\ell]}(z_t^{(\ell)})]), & \text{if } h \geq \ell. \end{cases} \tag{3.24}$$

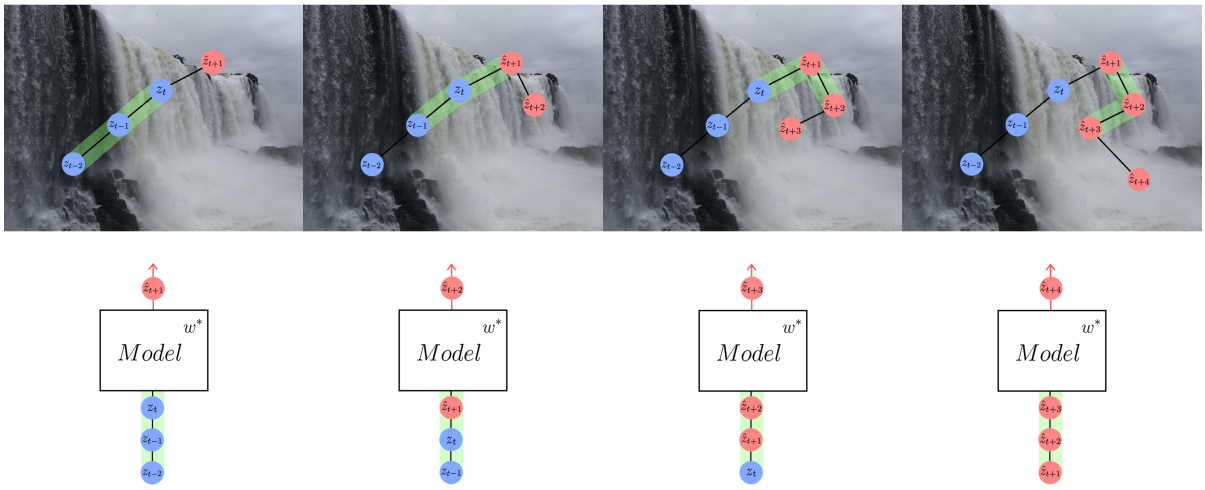

Figure 3.6: Recursive prediction 4-steps ahead with sequences of length $\ell = 3$, $z_t \in \mathbb{R}^2$. The first row shows how the scanpath predictions are drawn in the cascade image, whilst the second row shows how the model predicts (and is fed) recursively. Blue circles represent the available data at time t, the red ones the predictions and the filled green area are the input data for the model.

3.3.2 Direct forecast

The direct forecast aims to find the optimum parameters ω^* directly to the forecast horizon, so it is necessary to train a model for every time step ahead until reaching the horizon,

$$\hat{z}_{t+h} = f_{\omega_h^*}(z_t^{(\ell)}), \tag{3.25}$$

Note that every $\hat{z}_{t+h}$ must be obtained by fitting a different parametric function, mathematically this is analog to find the correspond weights of the h-th parametric function ω_h^*, for this process use equation (3.23) multiple times for every ω_h.

3.3.3 Forecast strategy selection

In this section we compare the recursive and direct forecasts strategies, specifically we focus on the case when f is nonlinear.

The recursive strategy necessarily forces an asymptotic bias due to accumulation of the one-step-ahead prediction errors [Brown and Mariano, 1984, Lin and Granger, 1994] where the direction of the bias will depend on whether the function is convex or concave [Granger and Newbold, 1976]. On the other hand, the direct strategy does not suffer from this problem. The latter strategy would be sensible if the model search were performed in a sufficiently flexible framework such as neural networks [Granger and Newbold, 1976].

Atiya and colleagues used both strategies and compared the MSE with general nonlinear models for two-step ahead forecasting [Atiya et al., 1999]. They find that the MSE when using direct strategy is asymptotically smaller than the MSE of the recursive strategy.

For the above, it seems to be that the best choice is the direct strategy, but note that doing this implies optimizing multiple models to generate the whole step ahead forecast sequence, that it could be impractical if generating the whole sequence is needed. On the same line, note that these factors are also dependent on the time series length ℓ, the forecast horizon h, and the model hyper-parameters.

Literature suggests that machine learning models provide better performance with direct forecasts and long time series. It is important to note that direct forecasts suffer from a large variance at long horizons with short time series. Empirically, the best performance has been found with machine learning models that have the ability to switch to linear models, such as neural networks [Taieb, 2014].

For instance, the main idea presented for selecting the forecasting strategy is to produce multi-step recursive forecasts, and then adjust the same architecture found before by using a direct strategy to train models at each horizon. This multistage strategy permits us to find a well-fitted architecture for making predictions, so then the models could be benefited as long as direct prediction models do not increase their variance too much. Empirical studies often found superior performance of the direct strategy compared to the recursive strategy.

To conclude, the direct strategy is often preferred over the recursive strategy since it avoids the

accumulation of errors. In particular, empirical studies often found superior performance of the direct strategy compared to the recursive strategy [Atiya et al., 1999, Kline, 2004, Sorjamaa et al., 2007, Hamzaçebi et al., 2009]. Perhaps, in the following sections we will first adjust models using the recursive strategy training with one-step-ahead targets, and then fit the models with a direct strategy trying to find better models at higher steps ahead.

3.4 Modelling uncertainty via MC-Dropout

Deep learning models have gained enormous attention in machine learning applications. Perhaps, these tools have not been able to capture model uncertainty per se. In 2016, Gal and Ghahramani develops a new theoretical framework casting dropout training in deep neural networks (NNs) as approximate Bayesian inference uncertainty [Gal and Ghahramani, 2016b]. Later, Gal and Ghahramani extend their work to recurrent neural networks (RNNs) by applying variational inference based on the dropout technique [Gal and Ghahramani, 2016a]. This new technique is called Monte Carlo Dropout (MC-Dropout), this can be interpreted as a Bayesian approximation of a Gaussian process.

Monte Carlo Dropout works by randomly removing units from the neural network during the training process, and then at inference time keeping it switched on in order to generate multiple different predictions of each instance. The average of these predictions is taken to be the final prediction and its variance is the model uncertainty.

For our purposes, modeling the uncertainty will help us to know how our neural model changes when we add some new feature, and to know which part of the scanpath prediction is affected by adding this new feature to the model. In the next chapter, we will see how adding features to models will make a noticeable difference between the uncertainty of the model without features and the model with new features.

3.4.1 Approximate Variational Inference

As we already mentioned, the above statement was extended for RNNs with the proposal of the recurrent dropout, which dropouts thought the recurrent connection that recurrent neural networks have. In this section, we review the background for making this uncertainty estimation via approximate variational inference (for more details check [Gal and Ghahramani, 2016a]).

Given training inputs $Z = \{z_1, ..., z_N\}$ and their respective outputs $Y = \{y_1, ..., y_N\}$ and a parametric function $y = f_\omega(z)$ of ω (for our purposes f could be a Multi-layer network defined in (3.3). Then, the likelihood distribution $p(y|z, \omega)$ for a regression task is defined as follows:

$$p(y = d|z, \omega) = \exp(f_\omega^d(z))/\sum_{d'} \exp(f_\omega^{d'}(z)). \tag{3.26}$$

With this, we can choose our prior $p(\omega)$ to be a standard Gaussian distribution (as is done in most neural networks initialization) to compute the posterior distribution over the space of parameters $p(\omega|Z, Y)$. Next, we can predict an output y^* for a new input point z^* by integrating

$$p(y^*|z^*, Z, Y) = \int p(y^*|z^*, \omega)p(\omega|Z, Y)d\omega. \tag{3.27}$$

Unfortunately, this posterior is not tractable in general, and we may use variational inference to approximate it with a distribution $q(\omega)$, and then minimize the KL divergence between the approximating distribution and the full posterior:

$$KL\left(q(\omega)||p(\omega|Z, Y)\right) \propto -\int q(\omega) \log p(Y|Z, \omega)d\omega + KL(q(\omega)||p(\omega)) \tag{3.28}$$

$$= -\sum_{i=1}^{N} \int q(\omega) \log p(y_i|f_\omega(z_i))d\omega + KL(q(\omega)||p(\omega)). \tag{3.29}$$

Next, the approximating distribution $q(\omega)$ for every weight matrix row w_k is defined as follows:

$$q(w_k) = p\mathcal{N}(w_k; 0, \sigma^2 I) + (1 - p)\mathcal{N}(w_k; m_k; \sigma^2 I) \tag{3.30}$$

with m_k is a variational parameter to be optimized[4], p the dropout probability and a small σ^2.

Finally, Gal and Ghahramani found that predictions can be approximated by replacing the posterior with $q(\omega)$ and solving using Monte Carlo integration:

$$p(y^*|z^*, Z, Y) \approx \int p(y^*|z^*, \omega)q(\omega)d\omega \approx \frac{1}{K}\sum_{k=1}^{K} p(y^*|z^*, \hat{w}_k) \tag{3.31}$$

with $\hat{w}_k \sim q(\omega)$. With this we note that this process is the same as performing dropout at test time and averaging results, concluding the MC-Dropout strategy.

[4]Note that m_k correspond to the RNNs weight matrices.

Chapter 4

Applications of Artificial Neural Networks for Scanpath Prediction

The Free Energy Principle (FEP) is based on the fact that self-organizing biological agents resist a tendency to disorder, hence they try to minimize the uncertainty (measured as entropy) of their sensory states [Friston, 2009]. This formulation implicitly means that the brain connections changes are a function of the pre-synaptic prediction and the post-synaptic prediction error [Friston, 2003, Friston, 2005]. In this sense, we pave the way for research on the modification of the nervous system, and thus to search this prediction error in the brain activity, this will be done in future work by disturbing the expectation of where subjects will look in a free-viewing task. For the above, it is our interest in this work to predict beforehand where people will look and where they pay attention.

The analysis of human visual attention is an active area of vision research, works revealed that human visual attention is modulated by bottom-up and top-down mechanisms [Itti, 2000, Connor et al., 2004]. The former mechanism operates fast and is in charge of shifting attention to salient visual features involuntarily, whilst the latter is focused on a longer-term cognitive strategy that changes our attention to task-related objects, therefore the top-down mechanism operates slower than the bottom-up mechanism. Furthermore, the two mechanisms are found to be independent [Pinto et al., 2013]. For a more detailed review check the Chapter 2.

In the field of computer vision, there is a wide variety of models that aim to mimic the cognitive process of visual attention. In recent years, an increasing number of saliency models have been proposed; some of them focus on the detection of salient objects, while others are concerned with predicting the fixations of the human eye, for more details see Section 3.1. Artificial Neural Networks have been emerged in the last years as the best predictive tool overall, reaching the state-of-art in many automated tasks such as regression analysis, classification, data processing, and robotics. For this reason, we presume that this architecture would allow us to predict the visual behavior of people while they are exploring a scene.

In the current chapter, we expose the different models we have created for scanpath prediction, we first start with the most naive model using recurrent neural networks which tries to predict the next position that subjects will see by taking only the previous scanpath information i.e the

past positions for predict the next ones. Then, we show our features exploration to improve our initial model performance. This by adding bottom-up factors as input to our model, such as the luminance contrast [Amano and Foster, 2014], the coefficients from a power-law fit of the power spectrum and the coefficients from a linear fit of the power spectrum [Piotrowski and Campbell, 1982]. We also tested more sophisticated features, like the last fully-connected layer of a pre-trained Convolutional Neural Network (CNN) called Mobilenetv2 [Sandler et al., 2018] or the saliency of the image obtained from a pre-trained CNN called SALICON [Huang et al., 2015]. Lastly, we decided to change our RNN module for one based on Attention [Vaswani et al., 2017], using as input a mix of features selected from the ones we saw earlier. We choose to use as input the previous scanpath and "foveated saliency images" which are foveated images obtained with some algorithm that fove an image (we use [Perry and Geisler, 2002]) and then it is passed through some saliency predictor model (we use the pre-trained SALICON model [Huang et al., 2015]). In addition, our model can carry out an algorithm known as Monte Carlo Dropout [Gal and Ghahramani, 2016a, Gal and Ghahramani, 2016b] that estimates the model uncertainty by finding the posterior probability of the position that a person will observe in an image.

4.1 Methods

4.1.1 Dataset

Our dataset contains data retrieved from 9 volunteers that during each trial they freely explored pictures while their eye movements were recorded. For every trial, the subjects explore 7 different image types, that are adjusted natural scenes (natural or NS) and 6 other control categories (inverted, pink noise, white noise, grey, black and white images. The whole dataset has a total of 46 natural images from the International Affective Picture System (IAPS) [Lang, 2005], and for every natural image, we constructed the 6 control categories mentioned before. The natural scenes were gamma corrected for the screen used, creating the adjusted natural scenes. The inverted images were the original natural images but upside down with the bottom part at the top of the image. The pink noise and white noise images were created considering the power spectrum and the phase of the natural images. The pink noise images had the same spectral power as NS but the phase is a scrambled version of the original NS. On the other hand, white noise images were obtained by flattening the power spectrum of the NS and keeping the same phase. Moreover, the grey, black and white images had the same average luminance as the original NS. The subject's eyes were 70 cm from the screen. The screen size was 1920×1080 pixels and the mean 32 pixels per cm was equivalent to 39.38 pixels per visual degree. Every image size was 1024×768 pixels and it was centered on the screen, the remaining borders were grey filled ($[173, 173, 173]$ RGB). The subjects performed the free exploration task with a total of 322 images in 30 minutes approx. whilst the eye-tracking data were retrieved with a sampling rate of 500 Hz. For all analyses of eye movements, we considered only data obtained from the left eye. All subjects were volunteers and accepted to participate through a written informed consent; the consent form and all experimental protocols were approved by the Ethics Committee for Research in Humans of the Faculty of Medicine from Universidad de Chile.

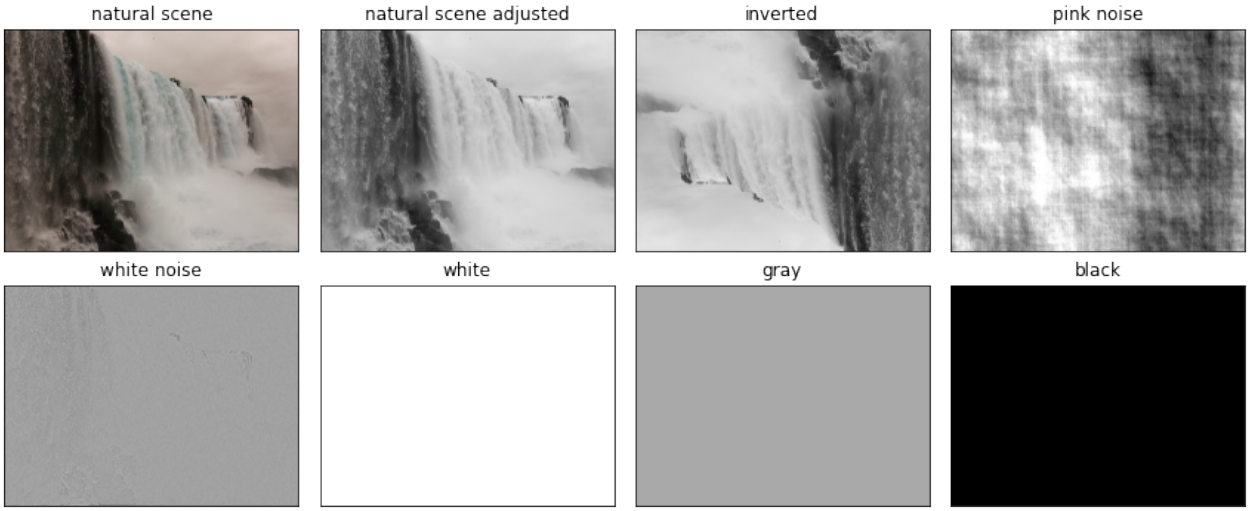

Figure 4.1: Different image types used for the free exploration task. Every natural scene is retrieved from the IAPS Database [Lang, 2005] and the others are modifications of it (the images displayed in this figure are not part of the IAPS Database).

4.1.2 Modelling Procedure

We aim to train different parametric models per subject and per type of image. In this way, we seek to predict subjects' visual behavior given a set of images types while the model was trained only with one image type. This will imply a predictive generalization i.e. the model has learned visual behavior independently of the type of image observed. In addition, this procedure helps us to compare how a given image type gives to the model information for predicting other image types. In Figure 4.2 are displayed some sampled scanpaths (8 seconds) from subjects during the experiment. The images on the left represent the scanpath of every subject when free-viewing a natural scene, the next two images represent the x-coordinate and y-coordinate position of the scanpath on time, the colormap changes as time increases.

For simplicity we cut the edge of every image, so the resolution change from 1920×1080 to 1024×768, we denote these data images as $\mathcal{D}$. Then, we split every trial into blocks of eye movements data with its respective image. Second, we choose a subject from the subject's space, $\mathcal{D}_S$, and an image type from the images space, $\mathcal{D}_I$. In other words, we are taking a sample from the set $\mathcal{D}_{SI}$ which is the inner join between $\mathcal{D}_S$ and $\mathcal{D}_I$ when the subject and image type are set. Then, we divide it into training (50% of $\mathcal{D}_{SI}$), validation (25% of $\mathcal{D}_{SI}$) and test (the last 25% of $\mathcal{D}_{SI}$) sets. The rest of the images categories and scanpath for the selected subject were concatenated to the testing set. Note that this process is repeated many times since we train a model for every image type and every subject. For instance, for every architecture, we train 63 models (9 subjects and 7 different image types)[1]. As we use neural networks, we standardize the data for stability purposes, besides we process the time series data by using a sequence length of 10 data-points which represents a subset of the person scanpath (i.e 20 milliseconds).

[1]Note that this is assuming a recursive strategy since if a direct strategy is used we have to consider one model for every desired step ahead.

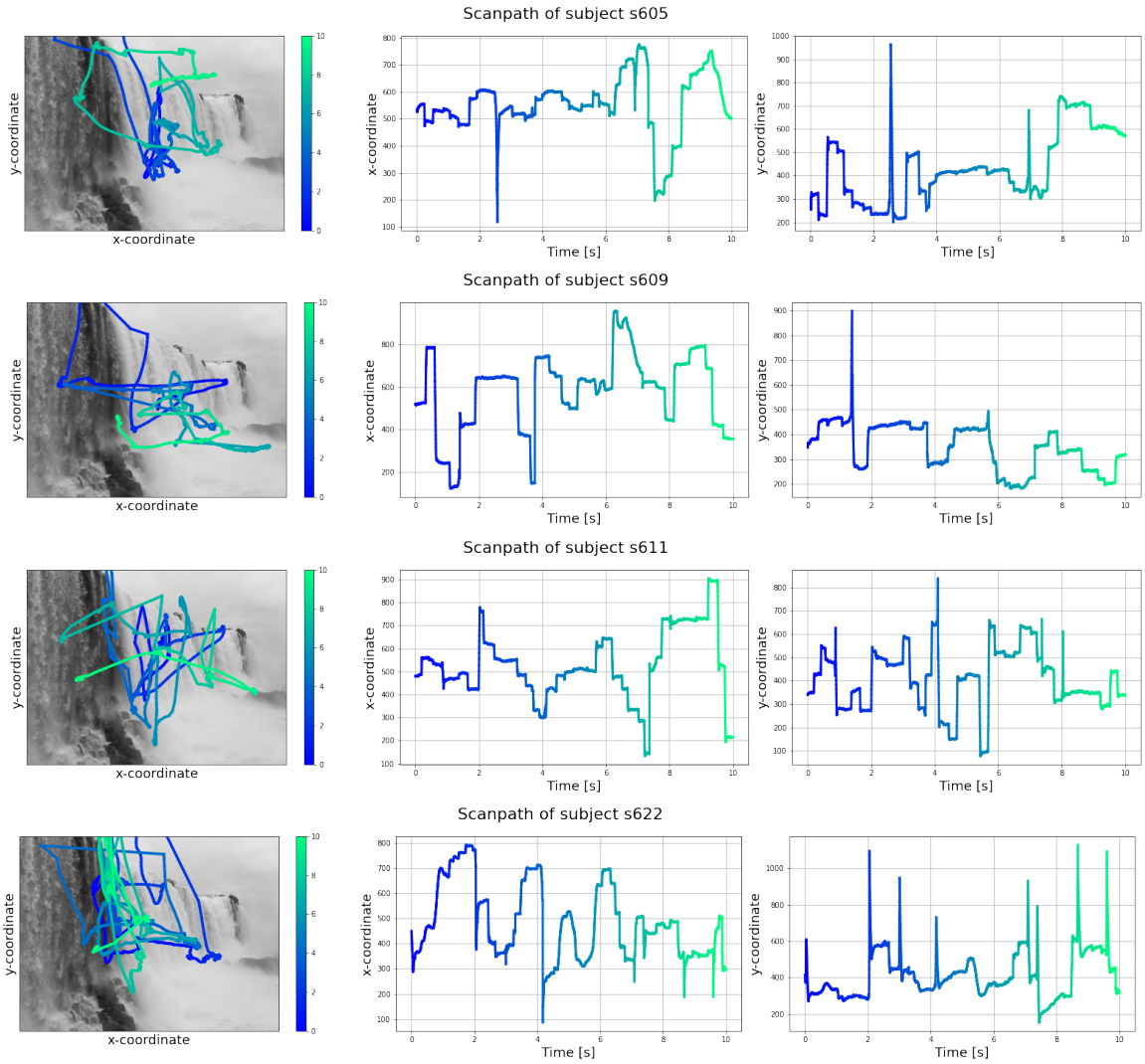

Figure 4.2: Scanpath from different subjects when free-viewing natural images (for simplicity the whole natural images data for a given subject was concatenated and displayed). The colormap changes as time increases.

The efficiency of the models are evaluated comparing between the ground truth and the predicted scanpaths, by using the next metrics: (1) the general scanpath amplitude error is measured by the MSE, (2) the time lag between the real scanpath and the predicted one is measured using the peak obtained from the cross-correlogram, (3) the ScanMatch metric [Cristino et al., 2010] is used for comparing the saccadic eye movement sequences spatially and temporally, (4) the MultiMatch algorithm [Jarodzka et al., 2010, Dewhurst et al., 2012] is used for getting general similarities between the real scanpath and the predicted, MultiMatch calculates 5 metrics that allow comparing scanpaths by shape, direction, length, position, and duration: (a) shape measures the difference between the shape of aligned saccades, (b) direction measures the angular difference between the aligned saccades, (c) length is the difference between the lengths of the saccades, (d) position is the euclidean distance between the aligned fixations, (e) duration is the difference in duration between the aligned fixations. For more details of the mentioned metrics refer to Section 2.8

Our analyzes were performed by grouping results by train image type and by predicted type of

image. Intuitively, the first case reflects how much "information" provides a given type of image for predicting the ocular behavior on other types of images, then the second shows how much "information" the other types of images have to predict the selected type of image.

4.2 Modelling scanpath with a recurrent neural model using positional information

Fixations and Saccadic events are not independent of each other when exploring an image. In fact, the scanpath past observations are directly correlated with its next positions where the people will fixate, for instance, our first strategy aims to take advantage of this.

The current section introduces a modeling schema that tries to learn this sequential process, for this we choose a Long Short Memory Term (LSTM) layer for this work. Section 4.2.1 further explains the architecture of the predictive model using as input only the positions obtained from the ocular scanpath in a free-viewing task.

4.2.1 Positional Scanpath and LSTM model

Firstly, it was thought of a way to model the selective order of people when free-viewing (see Section 2.5 for more details), for this a model based on recurrent neural networks emerges as a good decision for modeling this task, a recurrent model as an LSTM [Hochreiter and Schmidhuber, 1997] must be a good start point. For this, is necessary to set the length of the sequences aiming that these sequences contain the desirable spatial information from the ocular scanpath that allows the model to learn how to predict it, with this information it would be possible to adjust and predict the visual behavior of a subject. As Hu and collaborators show [Hu et al., 2020], in free-viewing conditions the temporal continuity performs well only within a short time interval lag of 100ms, so we aim to use this period for our sequence length, note that the temporal continuity is also known as the cross-correlogram [Box et al., 2015] which is used for checking the randomness in a data set i.e. as the temporal continuity becomes worse, the low auto-correlation in the scanpath will not give the necessary information for the model find an optimum that predicts the scan.

The Positional Scanpath model takes into consideration a window with the previous scanpath where the subject looked $z_t^{(\ell)}$, with this the model makes estimations about the future positions. This window is the time-series sequence of length ℓ where every time step corresponds to a position (of a pixel) in an image $z_t = (x_t, y_t)$.

Then, we feed to the model the previous subject data for minimizing the MSE error (see equation 3.23) between the ground truth next positions where the subject looked and the predicted ones. For the prediction of the next positions, we follow the recursive strategy (check equation 3.24) due to the greater amount of time it takes to use the direct strategy by having to train an exclusive model for each step ahead[2].

[2]For other models that we will see later if it will be worth using this strategy and just fit models to specific steps

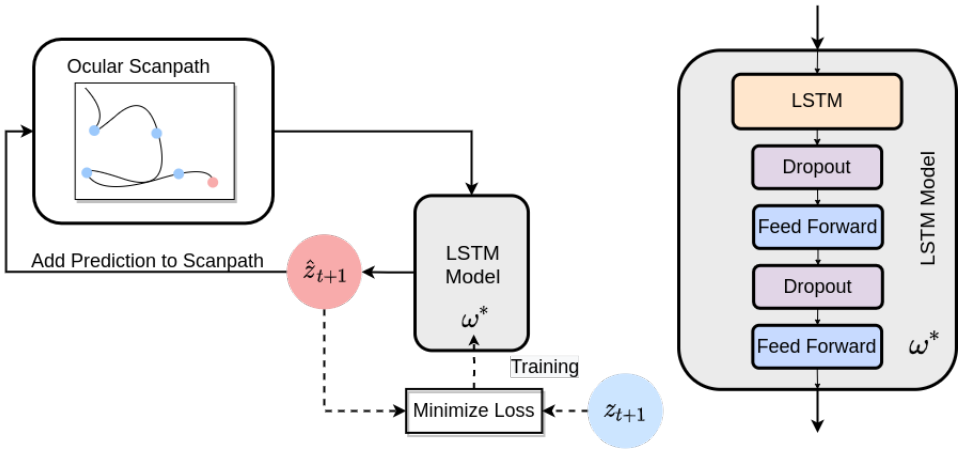

Figure 4.3: **PosScan architecture.** Prediction 1-step ahead with sequences of length $\ell = 5$. Note that Ocular Scanpath is a tensor with the scan-positions through time. Blue circles represent the available data at time t and the red one is the prediction.

Figure 4.3 show the process of how the positional scanpath and the LSTM model were used with a recursive strategy, in the diagram we see the last 4 positions seen by the subject $z_t^{(\ell)}$, with this a forward pass through the LSTM model is made obtaining the next position $\hat{z}_{t+1}$. If is on training phase then optimize its weights ω via minimization of the MSE loss between z_{t+1} and $\hat{z}_{t+1}$. Then, in order to get further predictions, we re-feed the LSTM model taking into account the earlier prediction $\hat{z}_{t+1}$ made, with this now the LSTM model input is $\hat{z}_{t+1}^{(\ell)} = [\hat{z}_{t+1}, z_t, z_{t-1}, ..., z_{t-\ell+1}]$, with a forward step again in the network we obtain the $\hat{z}_{t+2}$ prediction, then just repeat the same process until reach the desired steps ahead horizon h, for more details of the subsequent recursive predictions see Section 3.3.1.2.

In our PosScan model, we use an LSTM layer followed by Dropouts and Feed Forward layers. As hyper-parameters of our model, we use: 30 input units, a dropout rate of 0.2 and recurrent dropout of 0.2 with ReLU activation for the LSTM layer; Dropout layers has a rate of 0.2 to drop; Feed Forward layers has 20 and 100 units respectively with ReLU activations.

For our tests we vary the sequence length, we found that for training stability purposes and for achieving the lowest losses, the best value for the past positions (sequence length) is between 10 to 30 approximately (20 to 60 ms). Heuristically, we choose a sequence length of 10.

4.2.2 Analysis of PosScan model

We aim to measure both the spatial error and time lag of the PosScan model, for that literature suggest using metrics like MSE, ScanMatch, MultiMatch, and the peaks of the cross-correlogram inspired from biological processes which had to be measured (differences in neural spikes).

Sometimes is required to model the uncertainty of the future, to tackle this a Bayesian approach is needed, to do this its possible to add to our architecture the Monte Carlo Dropout algorithm [Gal and Ghahramani, 2016b, Gal and Ghahramani, 2016a], which finds the posterior probability of the positions where a person will see, given the previous positions seen. In Figure 4.4 the scanpath prediction of the subject s613 is shown, we can see how the predictions of the model get worse when

we increase the horizon steps ahead. Also, we can see how the uncertainty is greater in fixations than in saccades, this could be since the model did not have enough information about when the subject will perform a saccade while the input information is in a fixation, remembering that the model has as input only the scanpath last 20 milliseconds and the duration of the fixations is on average 250 ms, then the uncertainty increases in the fixations since the model does not know when the saccade may occur. This can be seen in the first column of the figure where the uncertainty is blurred at saccades and the fixations are sharpened, having a higher variance.

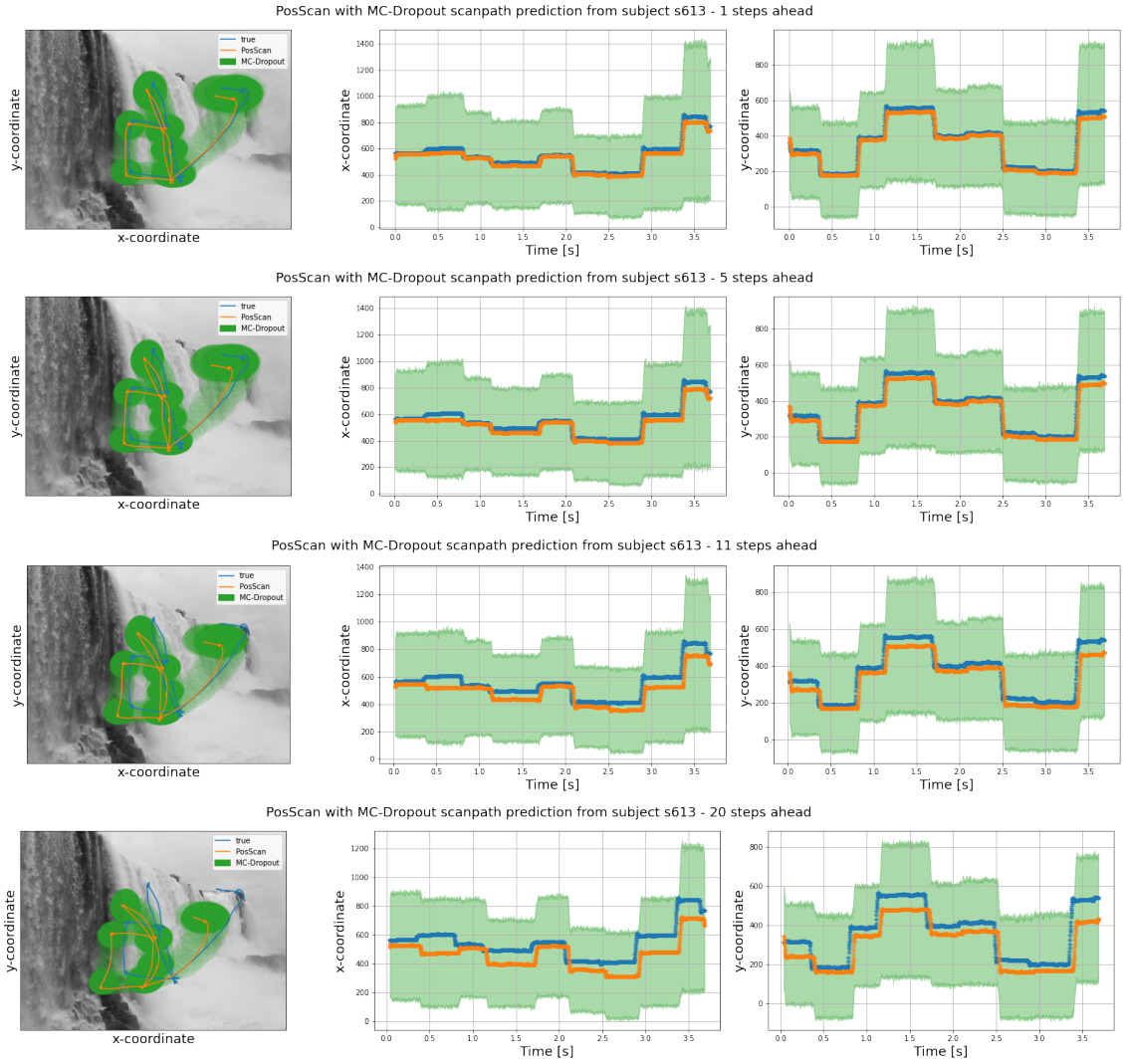

Figure 4.4: **The predictions get worse when we increase the horizon steps ahead and the uncertainty is greater in fixations than in saccades since the model did not have enough information about when the subject will perform a saccade while the subject is fixating.** PosScan scanpath prediction with MC-Dropout from subject s613 when free-viewing a natural image.

4.2.2.1 PosScan results grouped by train image type

We group the model results by train image type because we want to analyze if there are some differences in the prediction results among types of images. This grouping represents how much

"information" provides a given type of image for predicting the ocular behavior on all the other types of images. We found that there is a significant difference in the prediction results among the different types of images. Specifically, for the images visual content, we notice two groups in respect to the image types:

1. those with higher visual content refers to the images with many bottom-up features (or high-frequency components), such as natural, inverted, pink noise, and white noise images.
2. lower visual content images are those without frequency components, such as white, grey, and black images which have just one color.

The metrics which report the spatial error are the MSE, ScanMatch, the MM direction, the MM length, the MM position, and the MM shape. In general, the best performance achieved by the models is those with the image types that have the highest visual contents (i.e. natural, inverted, pink noise, and white noise types), while the results of image types without visual content (i.e. white, black and gray) are not good as the first ones. These statements can be seen in the Figure 4.5, where the MSE is lower and the ScanMatch is greater (reflecting good spatial predictive results) when using higher content images instead of the lower content images.

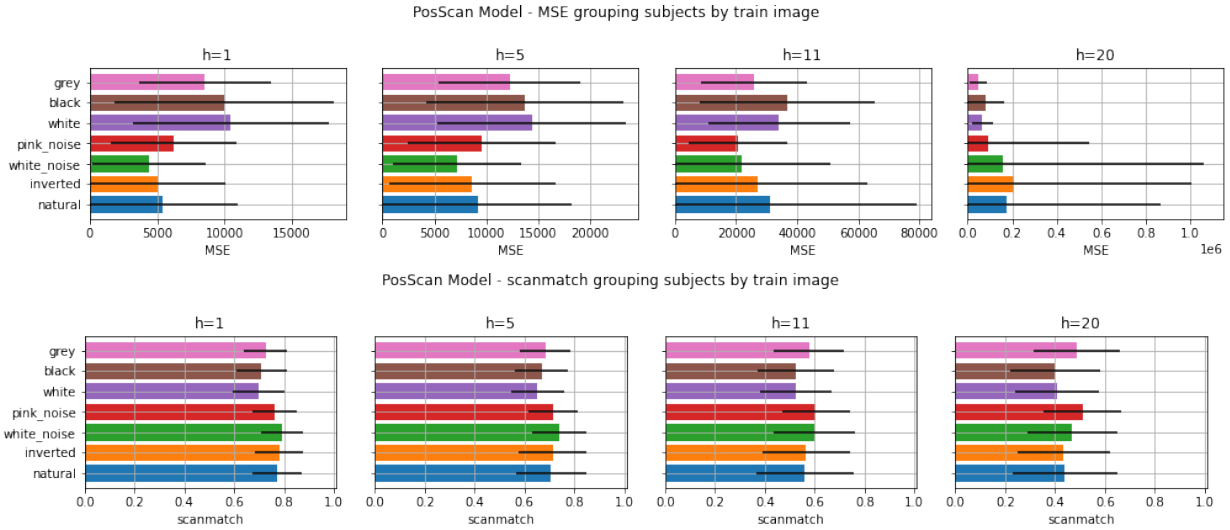

Figure 4.5: **Better performances on image types with higher visual contents than with lower visual contents, this is reflected in the MSE and ScanMatch, which are lower and greater in higher visual content image types, respectively. PosScan metrics results measured with MSE and ScanMatch, image types grouped by train image.** These metrics allow us to compare spatial differences between the predictions and the scanpath ground truth. Note that ScanMatch also measures the temporal phase error between scanpaths. A lower MSE represents a better prediction of the models, where 0 is it is lower bound. ScanMatch is measured between 0 and 1 when the latter indicates a better prediction.

We calculate the correlation between the image types given the metrics' distribution grouped by train image type, this allows us to see the two groups that are formed between images of high and low visual content, in Figure 4.6 the correlation matrix is reported and it shows high correlation within groups, but there is no relation inter groups i.e. correlations close to 0. Actually, when we increase the horizon prediction this group remains, it also seems that pink noise decreases its

correlation with respect to the high visual content group.

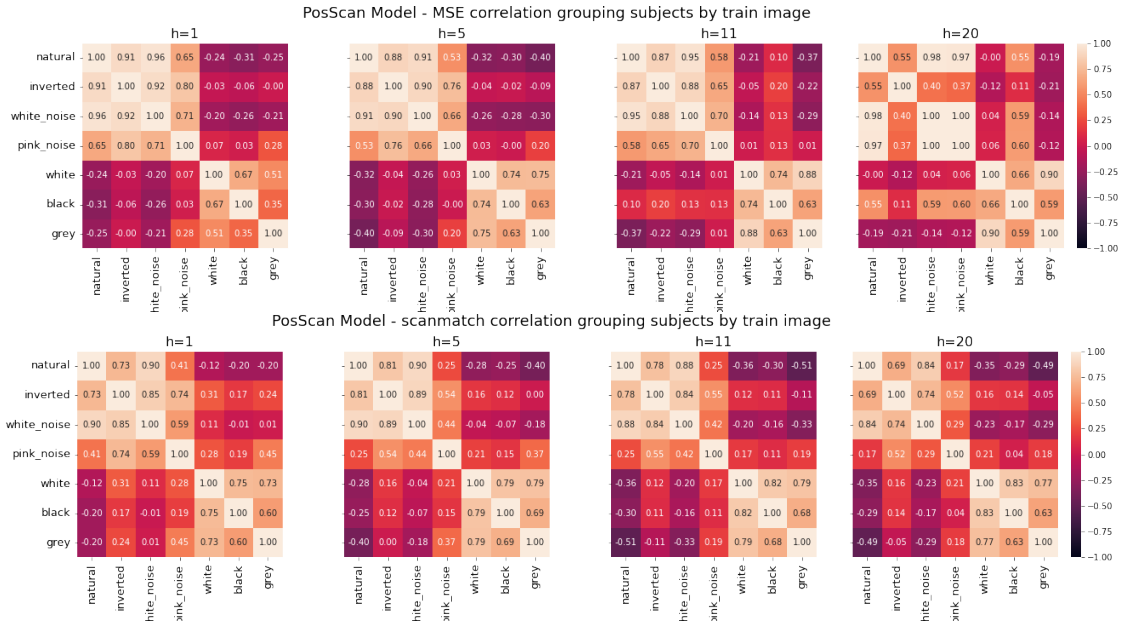

Figure 4.6: **Two groups can be seen in the correlation matrix between image types, these are the high and low visual content image types.** PosScan MSE and ScanMatch correlation matrices grouped by train image.

In addition, we also can notice the differentiation of the two groups in Figure 4.7 with the MultiMatch distributions. Note that MultiMatch has a low sensitivity when representing differences between one model and another as its values range from 0 to 1 but results are always above 0.8 approximately. We expect that this issue is related to our prediction strategy because we want to predict next time-steps so the scanpath differences between the predicted and the real ones might not be enough to be detected by this metric. However, considering the above it is also possible to establish certain differences (in this range of values) between the model predictions when we group by image type. About MultiMatch similarities in Figure 4.7 we see that:

- In MM shape is observed the distinction between the groups which has high and low visual contents, but in this case, pink noise shows a difference in the size of saccades more similar to the low visual content image types (white, black, and gray).
- In MM direction we see the differentiation between groups of higher and lower visual content without exceptions.
- In MM length same group differentiation by visual content, except for pink noise.
- In MM position we see the same group differentiation without exceptions.
- In MM duration, we can see a bi-modality in its distribution suggesting that for certain subjects the model correctly predicts the fixation durations by using as input only the past of the scanpath, but in other subjects, this information turns out to be insufficient to predict the temporal behavior of the scanpath.

Among the visual content mentioned above, the grouping pattern shown repeats again as we increase the horizon i.e. those with the highest visual content has their highest peak close to

0.8-0.9 indicating a better prediction in duration, while those with lower visual contents have their highest peak close to 0.4-0.5.

Besides, we can see that when we increase the horizon of prediction this bi-modality begins to collapse into a uni-modal distribution, even so, the visual content clusters remain the same, so the prediction of higher visual content images in duration is better than the duration predictions of the lower visual content image types.

The above grouping can also be seen in the correlation matrices, see Figure 4.8, it can be summarized in three cases (Figure reports just one metric for each case):

1. Pink noise changes from high to low visual content group (the first row in Figure with MM shape, MM length is in this group too).
2. Distinction between high and low visual contents (second row in Figure with MM position, MM position is in this group too).
3. The grouping distinction between high and low visual content begins to appear whilst we increase the horizon (third row MM duration).

When we increase the prediction horizon, MM shape presents highly correlation between all image type pairs triggering the disappearance of the visual content groups, and for MM duration this process happens the other way around (visual content groups begin to appear).

For the case of the MM shape, we speculate that this is due to the model predictions becoming unspecific to the image type. At lower horizons, the predictions are better for the higher visual content image types because its scanpaths have inherently more coherence to give information to the model about where the subject is going to fixate later. On the other hand, at higher horizons, the scanpath information is insufficient to make better predictions for the high visual content image types and this is how the model becomes unspecific to the type of image, and hence the correlations between all image types are high.

About MM duration, we can see that both visual content image types groups get worse as the horizon increases but at different rates. Note that the mode (peaks) of the low visual content distributions shifts toward left faster than the high visual content distributions.

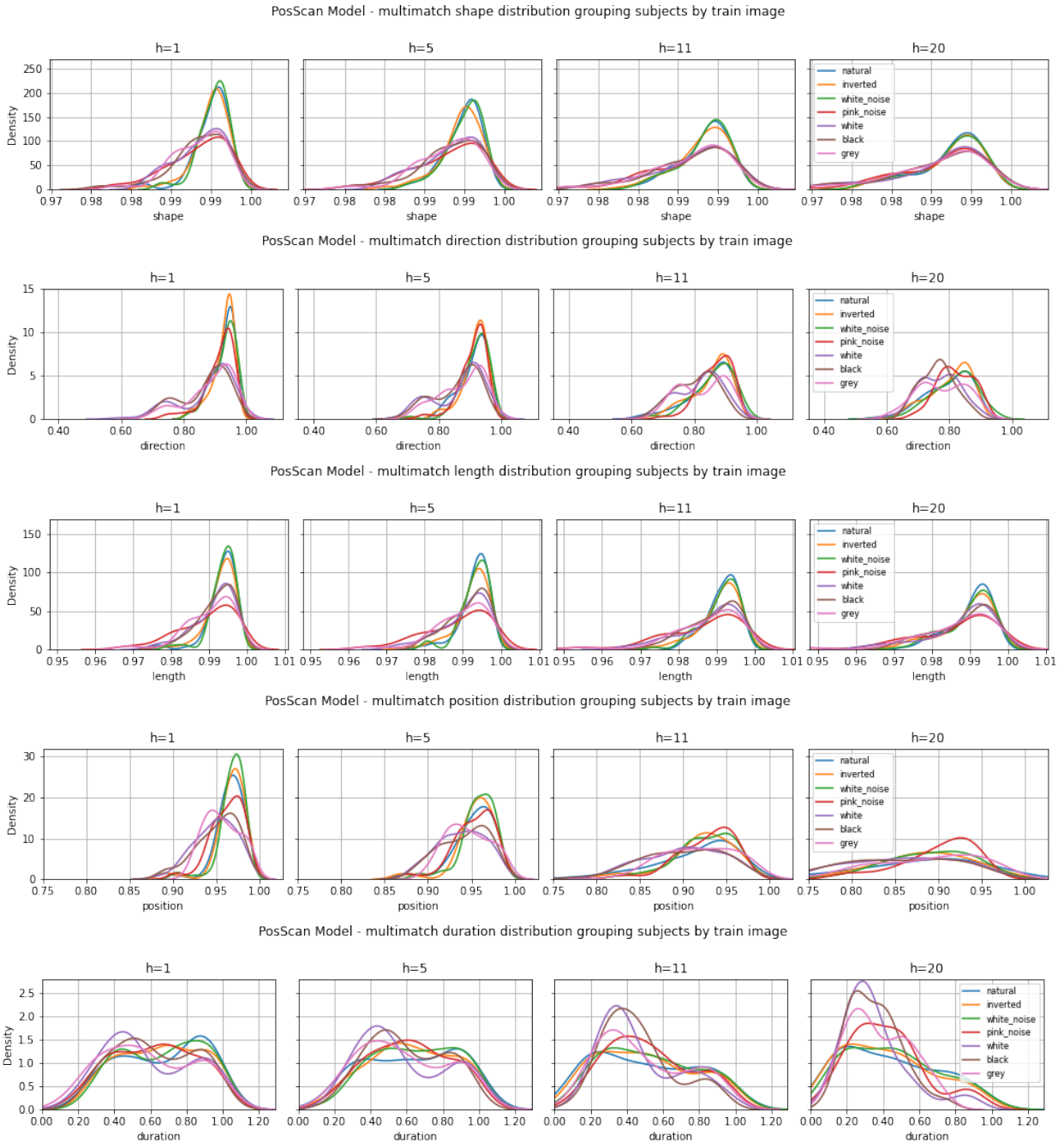

Figure 4.7: **The MultiMatch distributions show again the differentiation of the two visual content groups.** PosScan distribution results using MultiMatch metrics, image types grouped by train image. Note that the first four (rows) MM metrics measure the spatial features between the scanpaths (predicted and ground truth), while the last MM metric, duration, is the only one that measures temporal errors. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

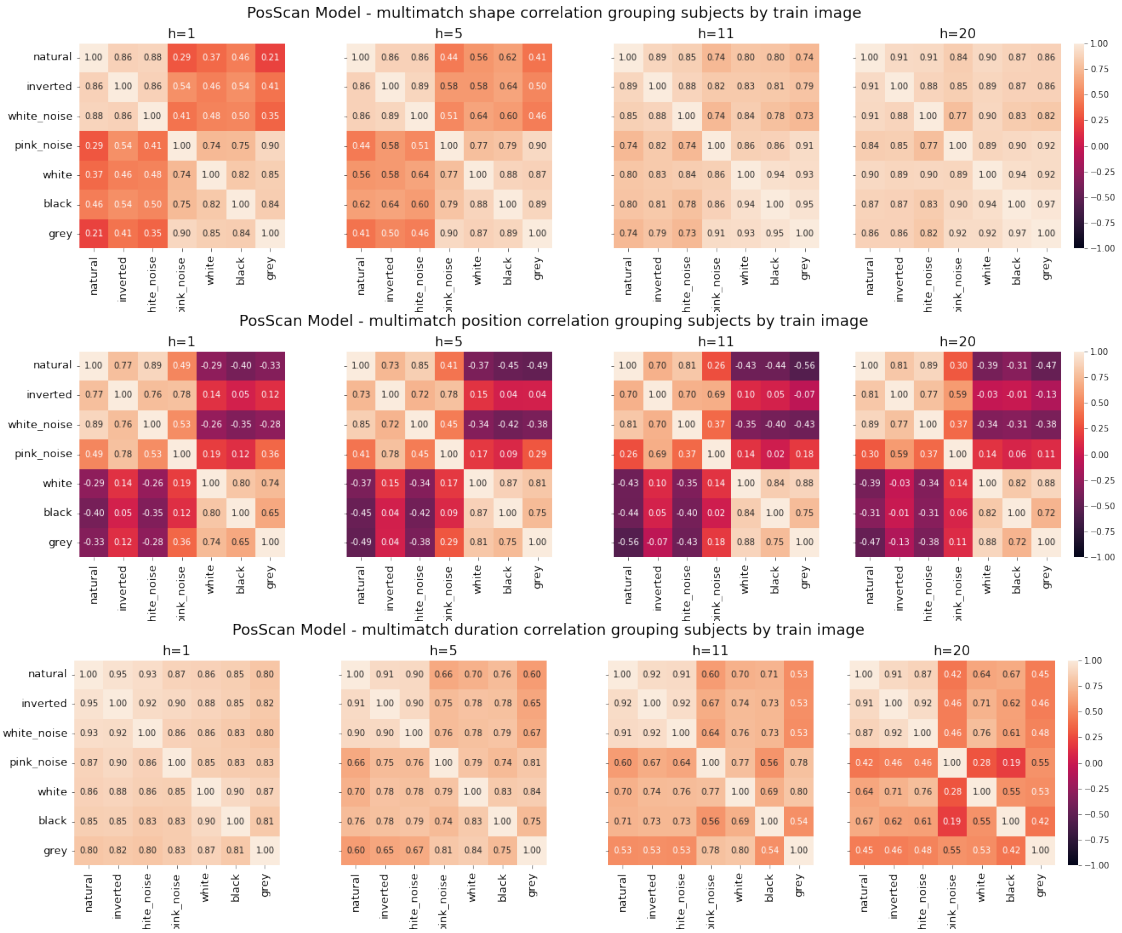

Figure 4.8: **Correlation matrices can be summarized in three cases of grouping: 1) pink noise changes from the high to the low visual content group, 2) clear difference between high and low visual contents, and 3) the grouping distinction between high and low visual content appear whilst the prediction horizon increases.** PosScan Multimatch shape, position, and duration correlation matrices grouped by train image.

Concerning the distribution of the cross-correlogram peaks, in Figure 4.9 we can analyze the temporal differences between scanpaths, it shows that for all image types the time lag (peak) becomes longer when we increase the prediction horizon, in particular, for the one-step-ahead prediction the predominant lag or peak is 5 samples approximately (10 milliseconds), then at 5 steps ahead the peak at distribution predominant it is on 7 samples approximately, at 11 steps ahead are 14 samples and at 20 steps are 21 samples of predominant time lag shift between the real scanpath and the predicted one.

Due to the multi-modality of the distribution of the cross-correlogram peaks, it becomes difficult to discern which model with its respective image type turns out to be better, see Figure 4.10, so we opt to show the cumulative distribution which allows us to select the percent of the data which corresponds to its time lag difference or cross-correlogram peak, so we can manipulate the operating point of the error i.e. for a given error we can find the amount of data that are under this error. For example, in the cumulative distribution if we select the 80% (0.8 in the graph) of the test data we find that the best models are those that intercept first when we trace a horizontal line through the x-axis (on 0.8). On the other hand, we can also think this backward by selecting

a time lag and intercepting it with the y-axis with a vertical line, then check which is the curve that has the highest percentage of data in this operation point. Thus, we can see that at 80% of data, the best performance measured in most of the cases is the pink noise results.

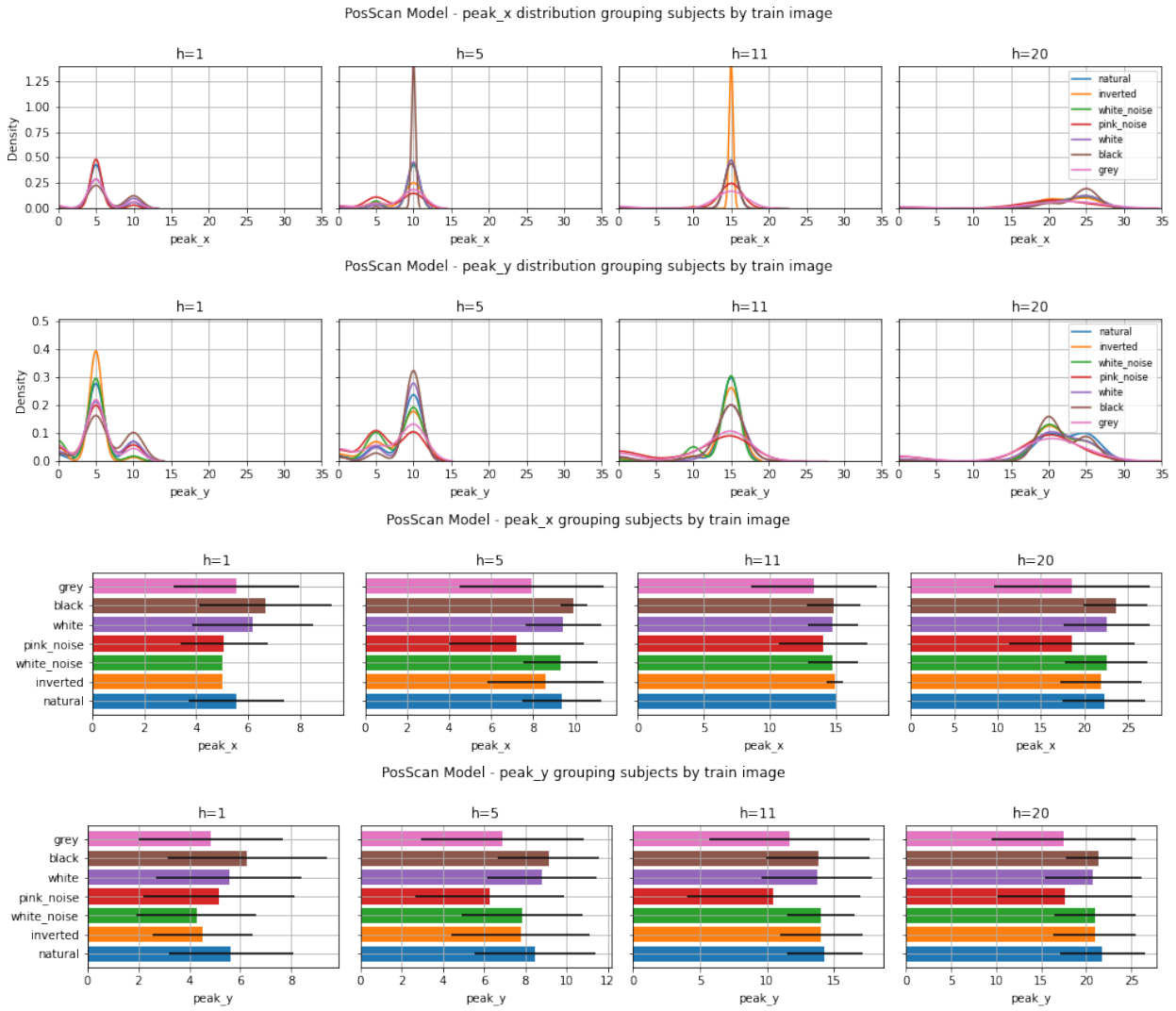

Figure 4.9: **The predominant time lag shift (peak) between the real scanpath and the predicted one becomes longer when we increase the prediction horizon. For the one-step-ahead prediction the predominant peak is** 5 **samples approximately (**10 **milliseconds), then at** 5 **steps ahead the predominant peak is on** 7 **samples approximately, at** 11 **steps ahead are** 14 **samples and at** 20 **steps are** 21 **samples.** PosScan cross-correlogram peaks results grouped by train image. The first two rows are the distribution of the cross-correlogram peaks and the latter two rows are a histogram that forces a Gaussian on the cross-correlogram peaks. Positive values indicate the prediction lag is after the ground truth and negative values, and negative values indicate that the prediction is before the ground truth.

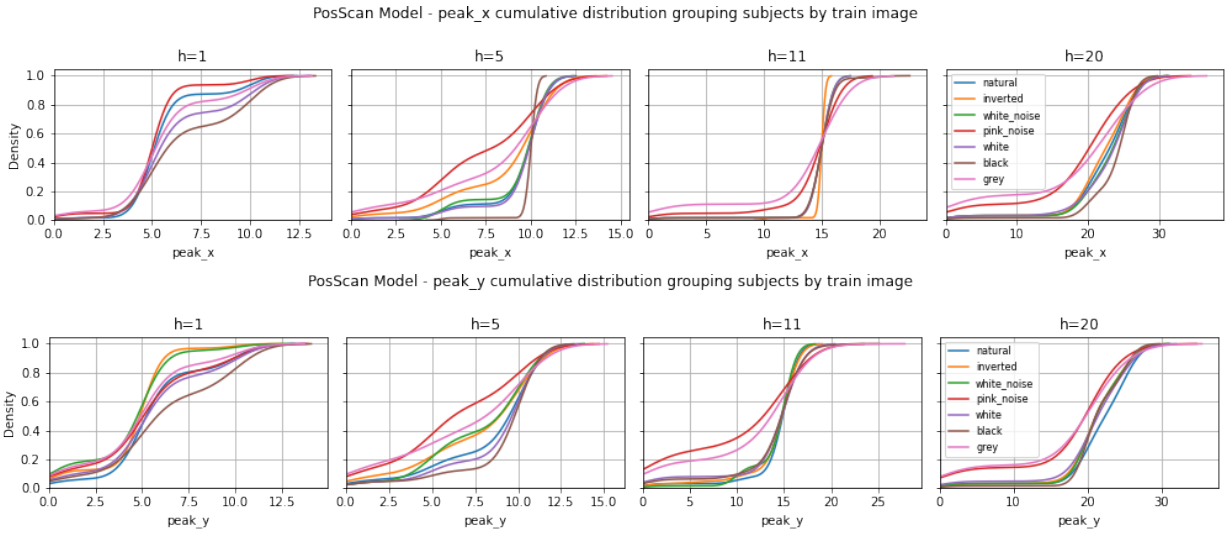

Figure 4.10: **The cumulative distribution allows us to select the percent of the data which corresponds to its time lag difference or cross-correlogram peak, so we can manipulate the operating point of the error according to our purposes.** PosScan cross-correlogram peaks cumulative distribution grouped by train image. The first row is the cumulative distribution of the peaks obtained from the cross-correlogram between the x-coordinates of the predicted and the ground truth scanpath and the second row is obtained the same way but using the y-coordinates instead. Positive values indicate the prediction lag is after the ground truth and negative values, and negative values indicate that the prediction is before the ground truth.

We report all the correlations image types combinations for every metric in Figure 4.11, note that these results are the same displayed in 4.6 and 4.8, but in this Figure we can see general trends between metrics and image types. The image types correlations between natural-inverted, natural-white noise, and inverted-white noise seems to be highly correlated (all correlations above 0.7) for every metric that was measured. Others moderately correlated pairs (majority of correlations above 0.7) are inverted-pink noise, white-black, white-grey, and black-grey. With this, we remark that the grouping by visual content is maintained across the metrics considering the above-correlated image types pairs.

Finally, as a summary, we want to emphasize that when grouping by type of training image we are seeing how each model benefits from the information provided by the scanpath that were obtained by observing a certain type of image. Hereby, we speculate that models fit better when the subjects' scanpaths are from free-viewing high visual content images, we believe that those kinds of images gives more information to the model in order to minimize the MSE error (and finds a better local-optima), and therefore, the models fit better its weights and make better predictions.

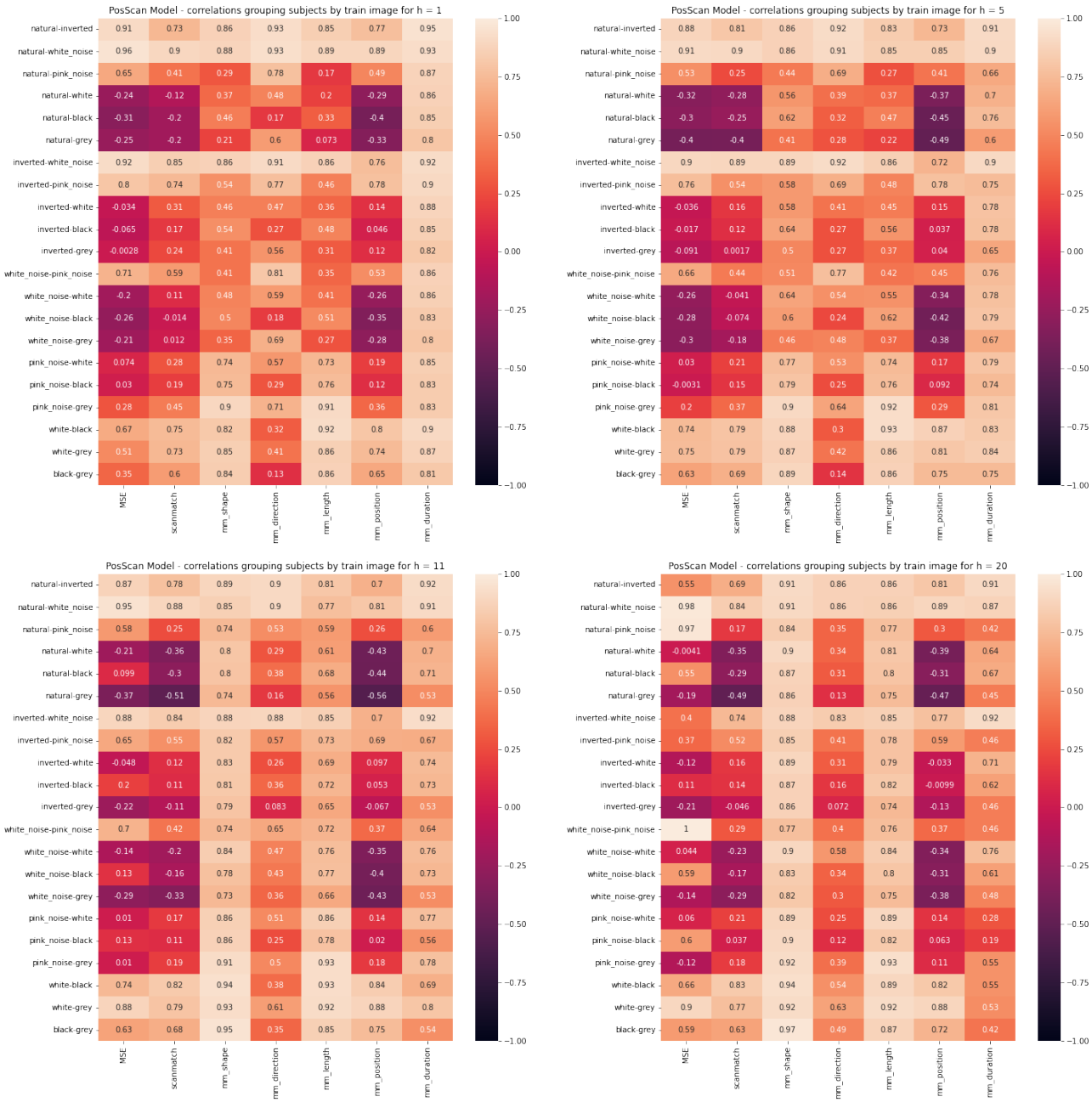

Figure 4.11: **The grouping by visual content image type is maintained across the metrics.** Summary with PosScan metrics correlation matrices (upper triangle) grouped by train image. Every value of the matrix is retrieved from the respective metric correlation matrix and its respective value given two image types, a column of the matrix has all the correlations with the image types combinations[3]. The first row shows correlations with prediction horizon 1 and 5, and the second row with prediction horizon 11 and 20.

4.2.2.2 PosScan results grouped by predicted image type

We want to measure if the input information to the models is sufficient for predicting the subjects' scanpaths, for that we group the metrics results by predicted image type (we take all the results of the models that predict a given type of image). The better performance of their respective group

means that it is easier to be predicted given the information provided when training with other groups of images.

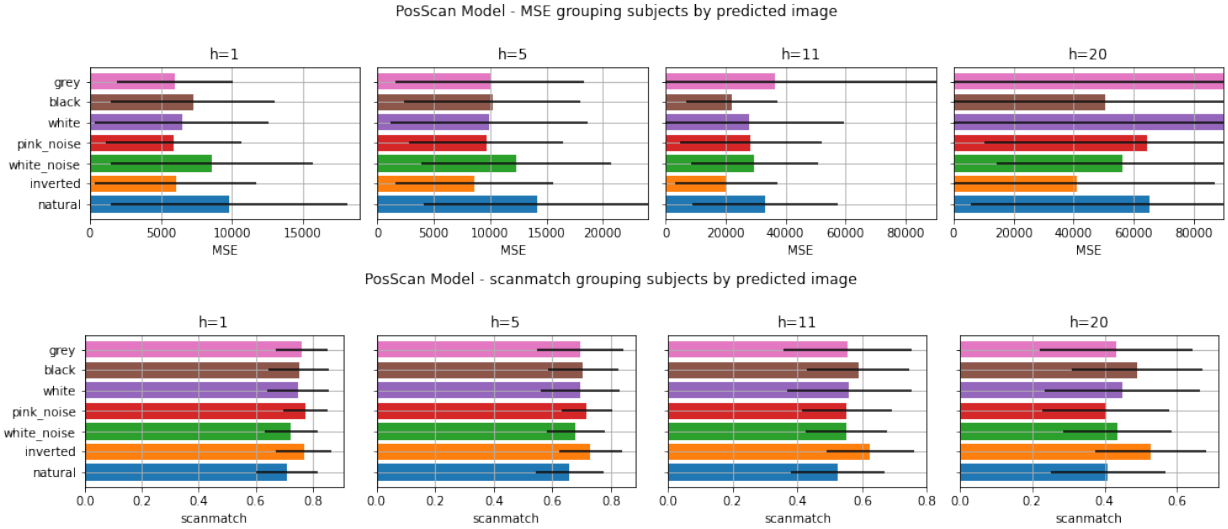

Figure 4.12: **MSE and ScanMatch show that is harder to predict when models try to infer scanpaths retrieved from natural and white noise images.** PosScan MSE and ScanMatch results grouped by predicted image. A lower MSE represents a better prediction of the models, where 0 is it is lower bound. ScanMatch is measured between 0 and 1 when the latter indicates a better prediction.

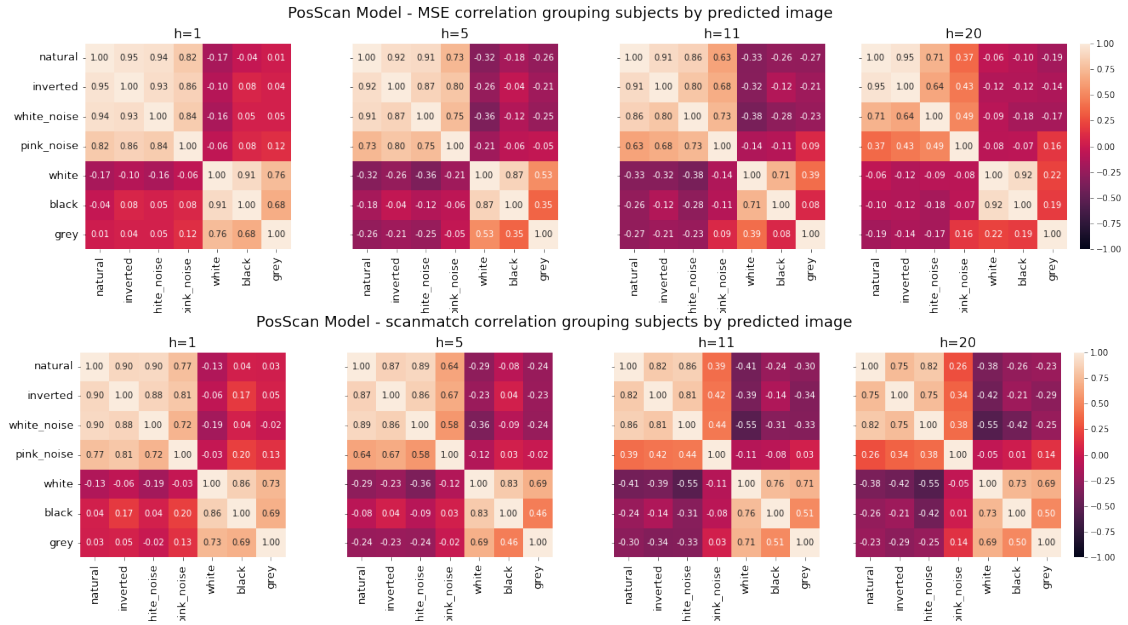

Figure 4.13: **The image type grouping between high and low visual content image types appears again.** PosScan MSE and ScanMatch correlation matrices grouped by predicted image.

The error in amplitude measured by MSE and ScanMatch in Figure 4.12 show that is harder for models to infer scanpaths on natural and white noise images. This means that the information given

from the scanpaths in other images types rather than natural or white noise is insufficient to model the scanpath behavior on these image types. When we calculate the correlations between them in Figure 4.13 we can see the visual content grouping clearly, we speculate that those scanpaths that were seen in images with lower visual content provide less information to the models when predicting image types with high visual contents.

In Figure 4.14 is shown the MultiMatch similarity when grouping by predicted image type:

- In MM shape those with lower visual content is better predicted, followed by those with intermediate visual contents (white noise and pink noise).
- In MM direction those with higher visual content can be predicted well when training with any other type of image, and those with lower visual content have slightly lower performance with respect to the higher visual contents group.
- In MM length those with lower visual content are better predicted.
- In MM position those with lower visual content are better predicted.
- In MM duration it occurs similarly as when grouping by training image, where we see a multi-modality in the distributions, the group that has the better duration performances (i.e. scanpath better predicted in duration) are represented by its peaks on higher MM durations, they are those with high visual content (natural, inverted and white noise), note that pink noise again changes from the group with higher content to the lower visual content group.

The peaks of the cross-correlogram distribution in Figure 4.15 show grouping between visual content image types, where in general the group with higher visual contents appear to be better predictable in time. This can be seen when we set a percent of data on the cumulative distributions, we see that those which reach lower peaks (meaning lower time shifts between the ground truth and predictions) by taking a greater amount of data are those scanpaths that were obtained from higher visual content images.

The same occurs in correlations, see Figure 4.16, where the group images with higher visual content have higher correlations between them, and the group with lower visual contents also have higher correlations between them. MM direction and MM duration metrics do not appear to have large differences across image types.

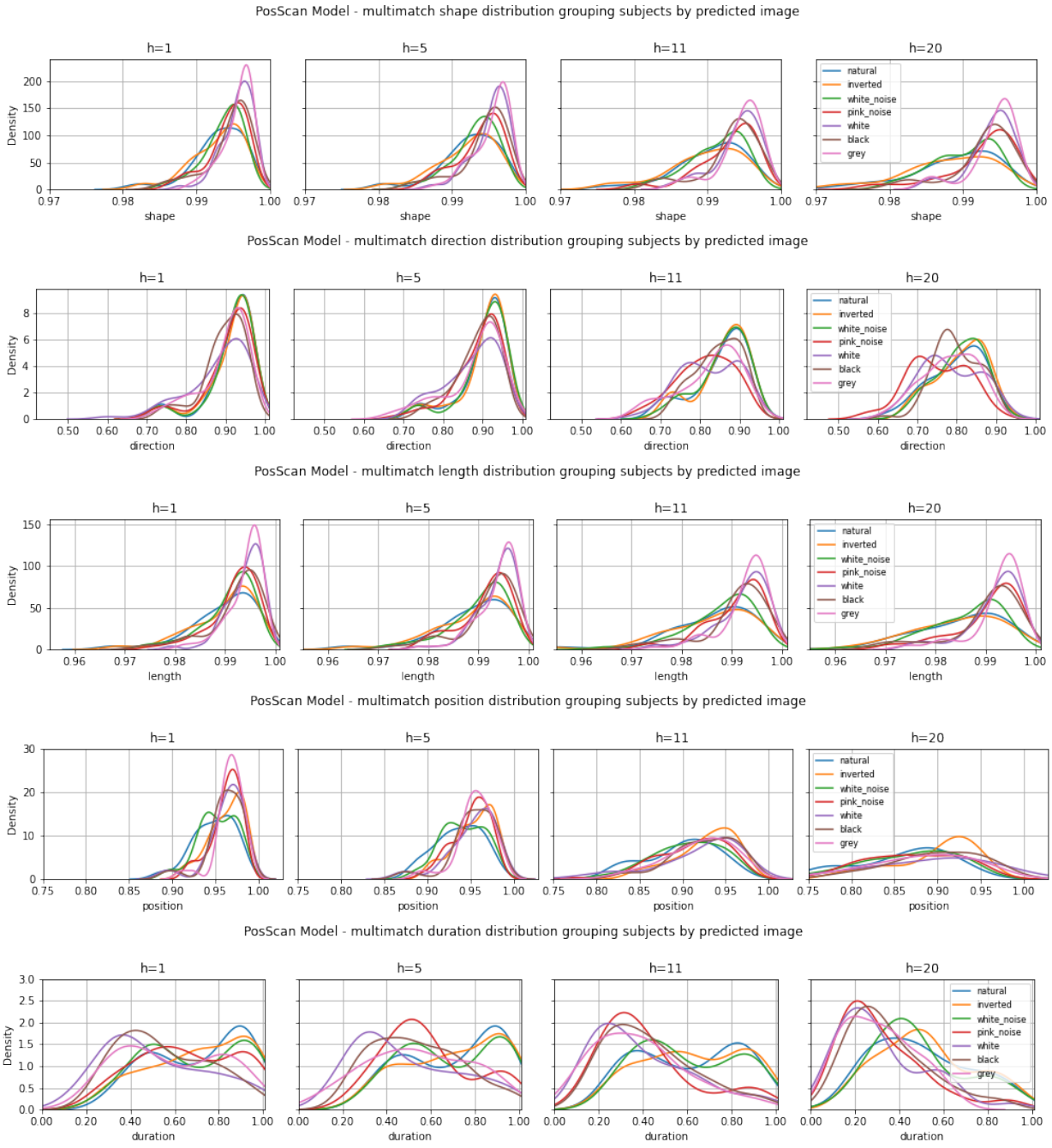

Figure 4.14: **In MM shape, MM length and MM position those with lower visual content are better predicted, in MM direction those with higher visual content can be well-predicted and lower visual content have slightly lower performance, and in MM duration there is a multi-modality in the distributions where the group that has the better duration performances are those with high visual content (except for pink noise).** PosScan distribution results using MultiMatch metrics, image types grouped by predicted image. Note that the first four (rows) MM metrics are in charge of measuring spatial features between the scanpaths (predicted and ground truth), and the last MM metric duration is the only one that measures temporal errors. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

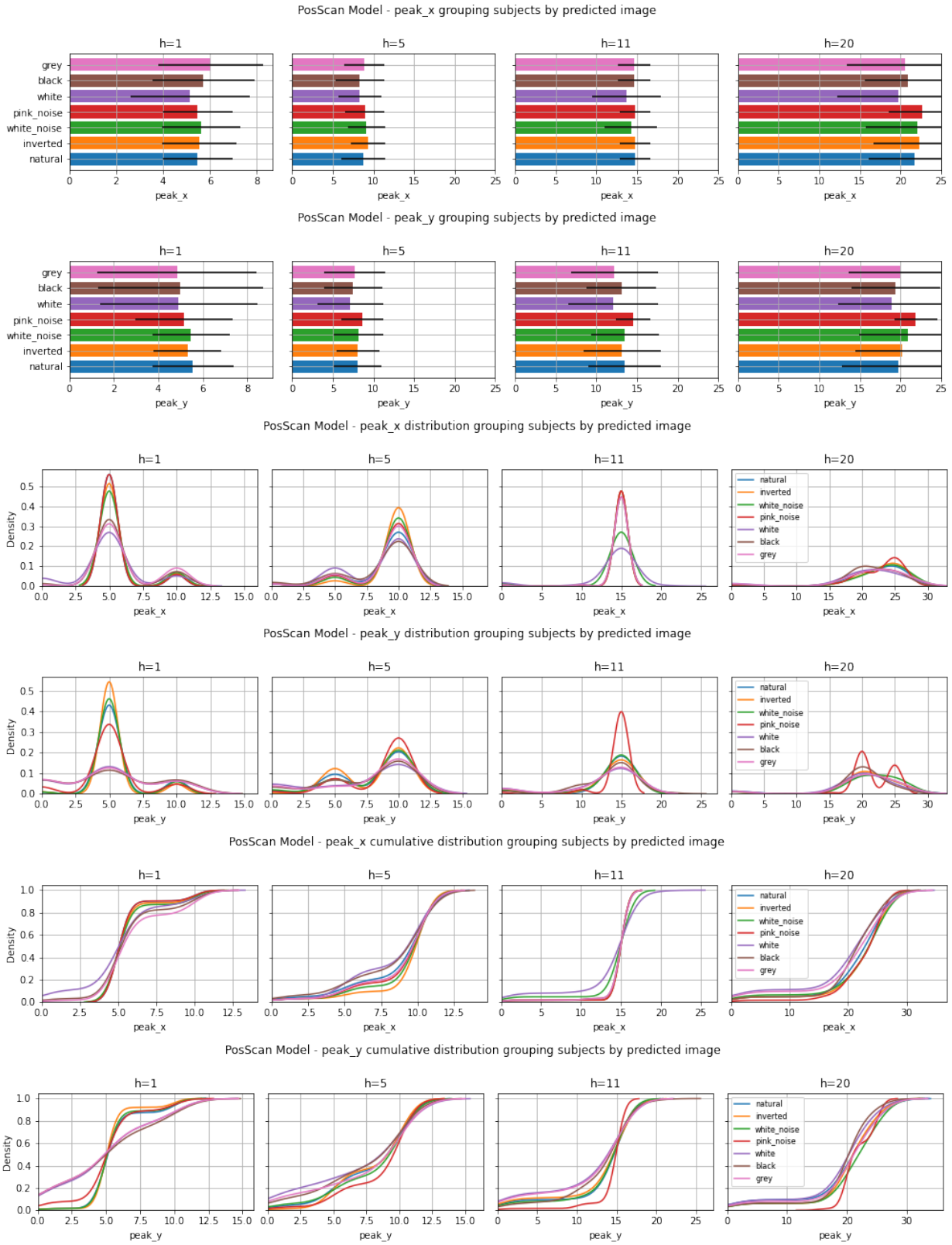

Figure 4.15: **The group with higher visual contents is better predictable while we increase the prediction horizon, by setting a percent of data on the cumulative distributions higher than 80%, the lowest time shifts (lowest peaks) results are those from higher visual content images.** PosScan cross-correlogram peaks results grouped by predicted image. The first two rows are the distribution of the cross-correlogram peaks and the latter two rows are the cumulative distribution of the cross-correlogram peaks.

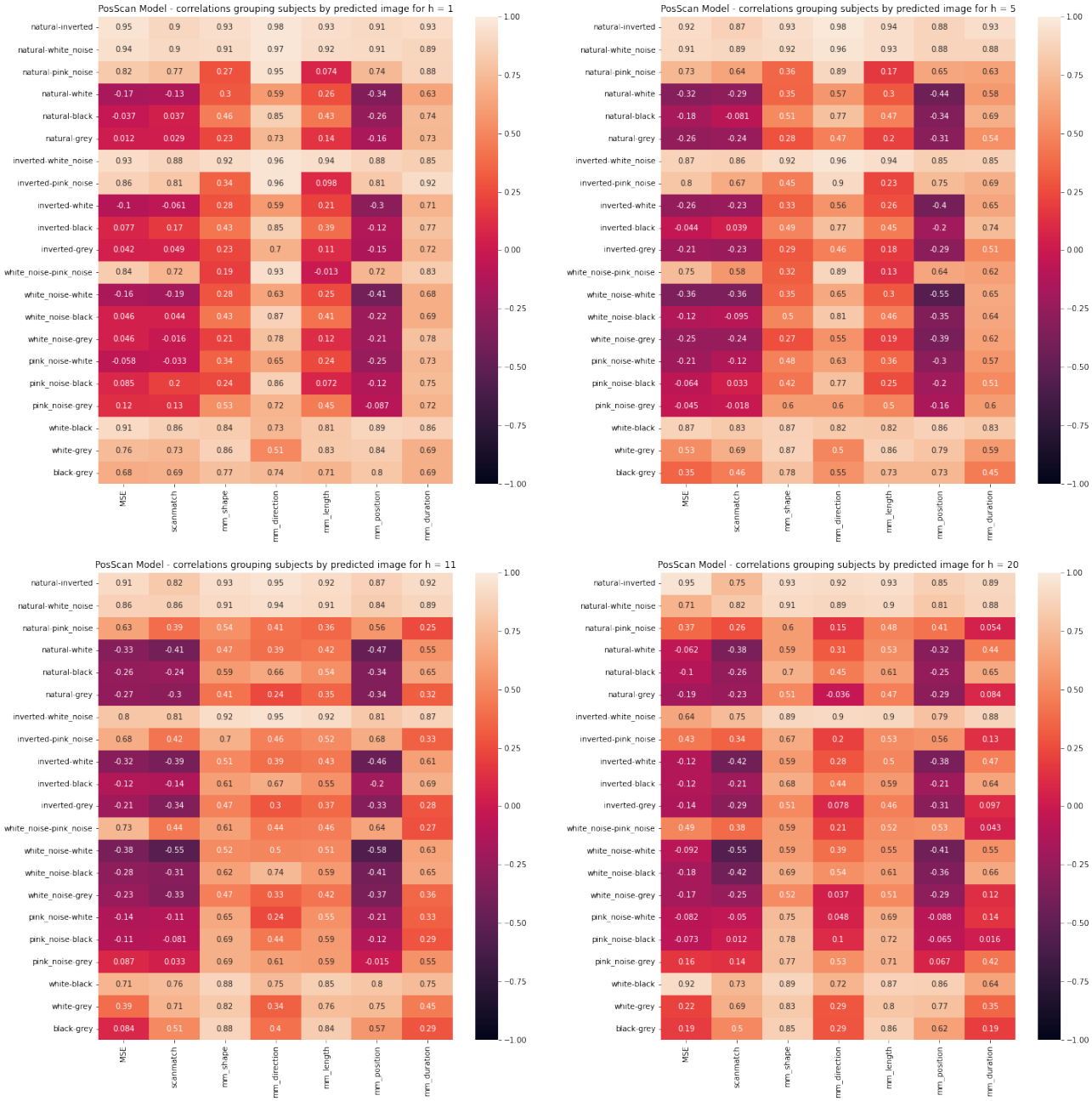

Figure 4.16: **The group images with higher visual content have higher correlations between them, and the group with lower visual content has higher correlations between them across all metrics.** Summary with PosScan metrics correlation matrices (upper triangle) grouped by predicted image. Every value of the matrix is retrieved from the respective metric correlation matrix and its respective value given two image types, a column of the matrix has all the correlations with the image types combinations[4]. The first row shows correlations with prediction horizon 1 and 5, and the second row with prediction horizon 11 and 20.

4.2.2.3 Prediction in other subjects rather than the trained one

We seek to compare the effect that the models have when they are trained with a certain subject but the predictions are made in other subjects. In this analysis, we focus only on the scanpaths

obtained from natural images, since as we have seen in the above sections, training with natural images helps models achieve better performances when they have to predict any other image type. Biologically, we see more often these kinds of scenes (natural) so we will pay more attention to this image type.

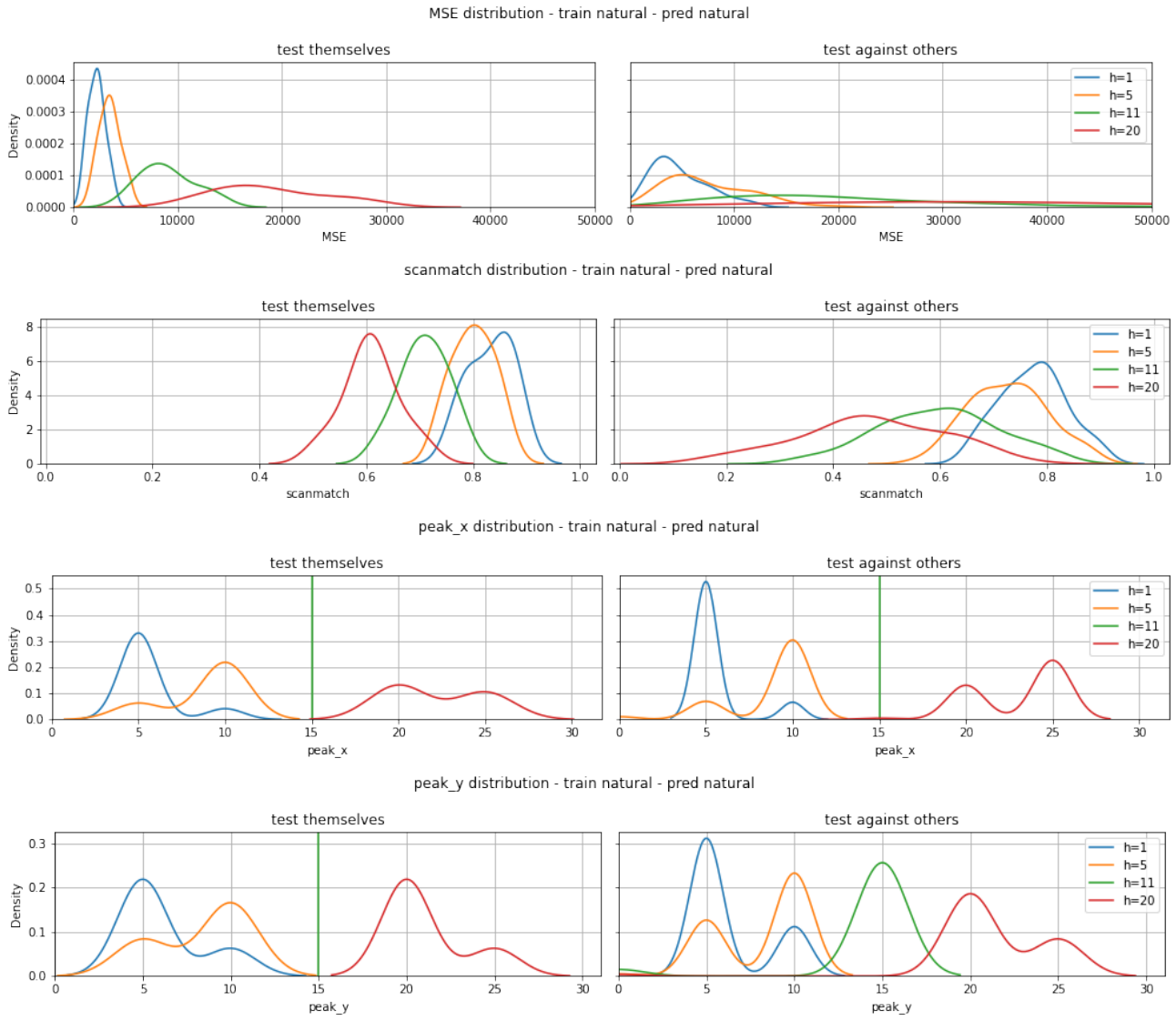

Figure 4.17: **The predictions are getting worse when we test against other subjects rather than the one on which the model was trained on.** PosScan MSE, ScanMatch, and cross-correlogram peaks distribution results, comparison between when model tested against the same subject and other subjects rather than the one the model was trained. The left predictions were made on the same subjects which the models were trained, the right predictions were made on all other subjects which are not the one which the model was trained.

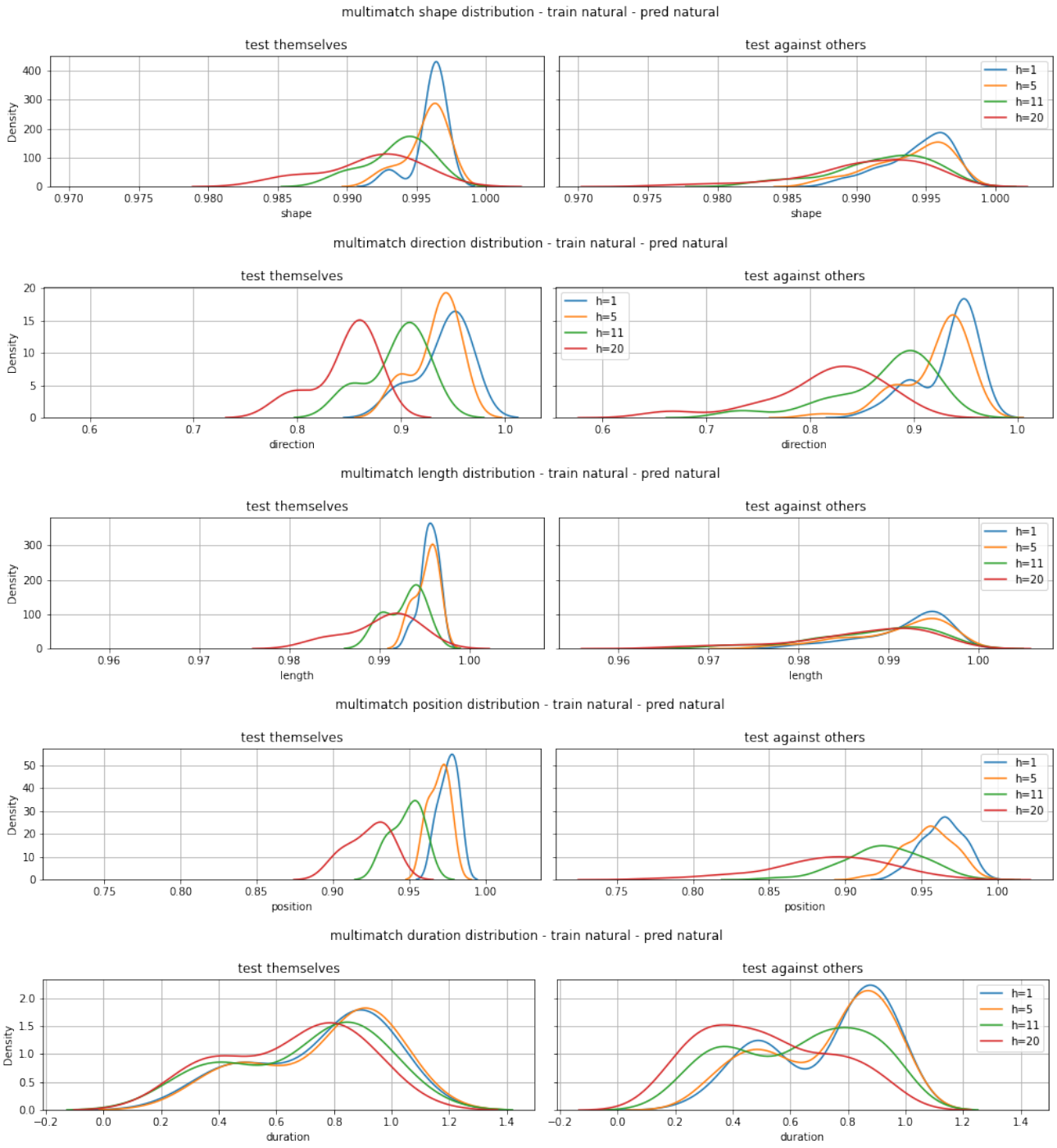

Figure 4.18: **The predictions are getting worse when we test against other subjects rather than the one on which the model was trained on.** MultiMatch distribution results, comparison between when model tested against the same subject and other subjects rather than the one the model was trained. The left predictions were made on the same subjects which the models were trained, the right predictions were made on all other subjects which are not the one which the model was trained. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

In Figures, 4.17 and 4.18 we report the distribution metrics for different prediction horizons (1,

5, 11 and 20), the left predictions were made on the same subjects which the models were trained, the right predictions were made on all other subjects which are not the one which the model was trained. On all metrics, we see how the predictions are getting worse when we test against other subjects rather than the one on which the model was trained on.

We also check how are some specific scanpaths, and although we take just one of many models (that we could have been taken), we think that it reflects in general how the predictions are when we trained and testing on the same subject, and when it was trained with one subject and testing on another one. In Figure 4.19 we can see the scanpath prediction made by two different models, the first one (first row) was trained with natural images data retrieved from the subject $s607$, and the second one (second row) was trained with natural images data retrieved from the subject $s609$, both models made its predictions using data from the subject $s617$. The rows on the Figure represent the x and y coordinates of the scanpaths (ground truth and predictions) for the two models. If we see the first (or the second) column we can compare the x-coordinates (or the y-coordinates) predictions [5], the predictions are worse when we test on another subject rather than the one which we retrieve the data for train the model. The fact that models cannot generalize correctly for other subjects is because the models did not have information to learn the specifics subject patterns, this suggests that every subject holds their exploring self-pattern.

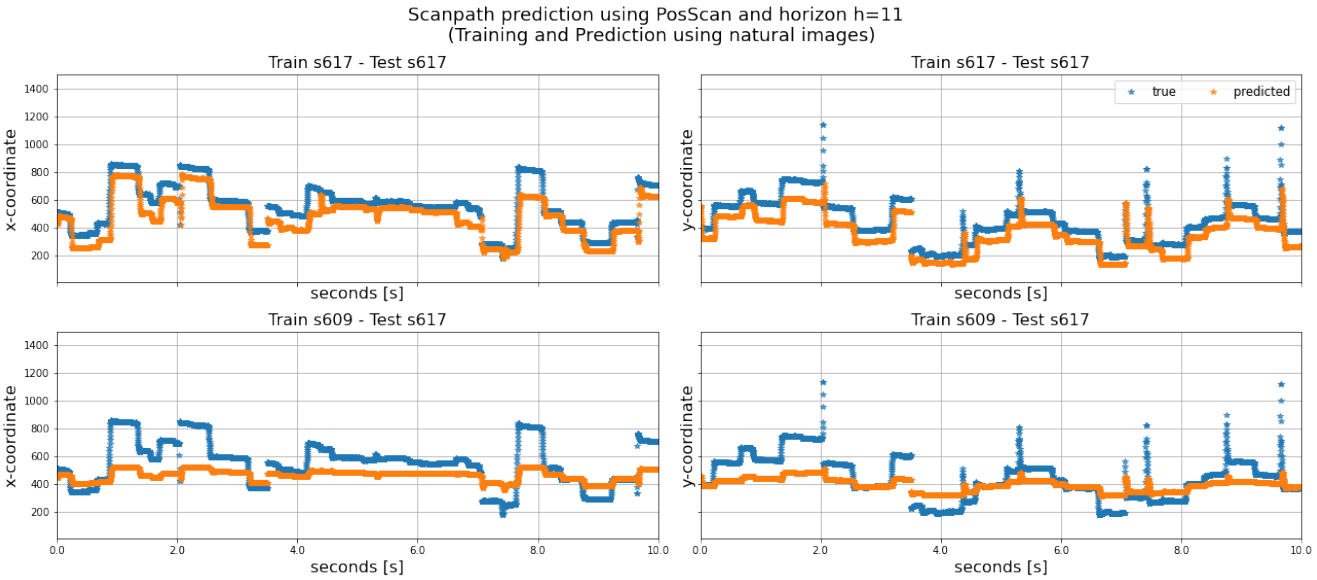

Figure 4.19: **The predictions are worse when we test on another subject rather than the one which we retrieve the data for train the model.** PosScan scanpath predictions comparison when trained in two different subjects ($s617$ and $s609$) but testing only on one of these subjects ($s617$).

4.2.3 Remarks

Our PosScan architecture shows that with the ocular scanpath is possible to fit models for making predictions of subjects' scanpaths. The most important fact is how models' performances are better when predicting on higher visual content image types than on those with lower visual content, this

[5]Note that the ground truth is the same for every column because we tested on the same data, subject $s617$.

implies that ocular scanpaths are more affected (by changing subjects' attention) when they free-view higher visual content than the lower content image types. There could be two explanations for this, the first one is that models trained with higher visual content images are able to capture more information about the environment since these kinds of images guide the subjects to inevitably pay attention to specific zones. The second one is that scanpaths from lower visual content images have more random components, and so subjects' scanpaths are guided only from their attentive system without any externals, so models trained with this kind of images are not able to capture the information about the environment.

Unfortunately, PosScan models cannot infer when subjects will going to perform a saccade whilst they are fixating, instead they can only make a better prediction after the beginning of a saccade.

In addition, we tested the models on other subjects rather than the one which they were trained on, resulting in worse performances. These results are consistent since people have different ways to explore new environments, so a model that never saw a scanpath from another subject did not capture the specific subject scanpath patterns.

4.3 Selecting features to enhance the model

In accordance with the above sections, solutions are thought to reduce the time shift error between the scanpaths predictions and ground truth, because until now the models have not been able to predict saccades but only follow the scanpath with greater or lesser delays (depending on the image type that the subject saw), this delays can be seen at the peaks of the cross-correlogram.

We detect that our models could have failed due to fact that they have only a portion of the scanpath past for making predictions, in fact as input we give only 10 data points of the past representing only 20 milliseconds. The literature shows that fixations duration has a minimum pause time of the eye without any stimulus processing average of 200 milliseconds [Salthouse and Ellis, 1980]. Salthouse and colleagues also found that there is duration concerned with stimulus processing which assumes a duration between 50 to 100 milliseconds. With this in mind, we need to take into account for our models that fixations could last around 300 milliseconds.

One idea was to increase the size of the sequence delivered as input to the model when trained, and thus have more information that fully characterizes a fixation so the model could infer if the fixation has already been a long time and the subject will have to perform a saccade. One of the disadvantages of this approach is the great computational cost involved, expensive time to train and test models, and data loss at the beginning equal to the size of the chosen length of the sequences. In the tests we performed, models could not find an optimal solution that minimize the error underestimating the amplitude of saccades in most of the cases, in fact when we tried to train for sequences of length greater than 50 the validation loss increased so much forcing the models to early stop.

Another idea was to add a Hazard function which models the probability that a saccade will be done given the time when the previous saccade was made, unfortunately, we could not implement this since we focus our attention on other tests, so we leave it as future work.

We thought about features to add as input so the model could improve, from the biological point of view it could be added the binocular vision data (right eye data) as a new feature for models, remembering that PosScan only uses data from subjects' left eye. Further, it could be added features like the pupil diameter since the literature shows that local pupil luminance responses at the location being prepared for an upcoming saccade, suggesting that pupil size is modulated by the luminance level at the location selected by spatial attention [Wang et al., 2018]. However, there is a problem if we want to add the pupil size, is that it is a variable that does not depend directly on the spatial position observed, so it would be necessary for the model to predict the pupil size if we use the recursive h-step ahead prediction approach, for that we decided not to try this features at least on this work.

Following with biological features that we could add, as we discuss in the Sections 2.3 and 2.4 subjects decide where to look influenced by bottom-up and top-down factors, then we decide to test these kinds of features on the models and check if they improve. We made many tests with scanpaths and features retrieved from free-viewing on natural images, so we train and test on this image type. The model training and tests were made only on 3 subjects ($s605$, $s620$ and $s622$), this gives us an insight into which feature could improve the model.

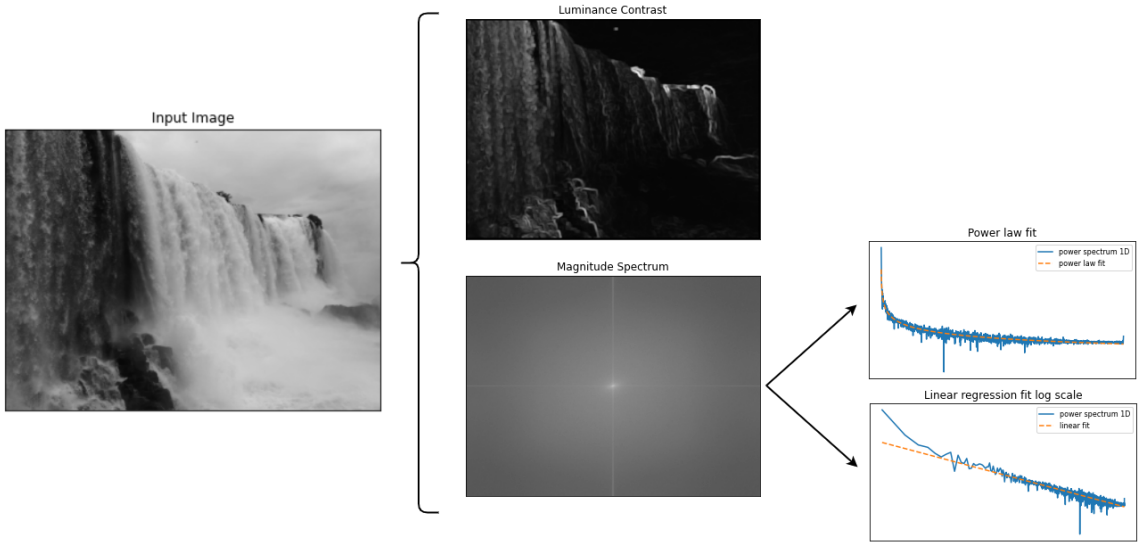

Figure 4.20: Calculation example of Luminance Contrast (LC), Power spectrum (PS), and fitted curves from the PS.

We test adding the Luminance Contrast (LC) to our PosScan model as a local spatial feature, with this the model has information of where and what the subject is seeing. In parallel with the above, we test other features that were retrieved from the Image Power Spectrum (I-PS) extracting the two PS's principal directions on images. One idea was to fit a linear regression on the I-PS in log-scale and give it as input to the model, we test by adding both and separately the slope and the positional coefficient[6]. Another idea was to fit a power law curve to the I-PS and use its proportionality and exponential coefficients, in Figure 4.20 the features extractions process is shown. As mentioned above, features take into account only the global characteristics of images, we perform the same process but now by selecting just a boundary of where the subject is seeing,

[6]This is also known as the signature of the power law.

and use the same process for extracting features. We test this by calculating the slopes of the local I-PS with windows of 40×40 pixels. Also, we test adding to the model a combination of the above features i.e. add the Global and the Local slopes (GL)[7] from the I-PS.

In Figures 4.21,4.22, and 4.23 we can see the results for different models and realize that the ones which has biologically inspired features perform better when we compare them against our PosScan baseline model. These improvements can be seen in almost every metric, only in a few, this is not observed like MM shape, MM length, and MM position because our baseline model (PosScan) already achieved above 0.9 in these metrics, so improving it more than that is more difficult by simply minimizing the MSE loss between the scanpath ground truth and the predictions.

Results show that adding or not the positional coefficient (see in Figure the comparison between *test_features_slopes* and *test_features_lin*) results remains the same for all metrics. Note that this assumption just applies to natural images, actually we tested on other image types with low visual contents like grey, black, and white and the prediction performance worsened.

Next, we note that results are pretty close between them when adding the power-law or the linear fit as features (see in Figure the comparison between *test_features_pl* and *test_features_lin*, respectively), this could be due to the information embedded in these two features are almost the same.

Figure 4.21: **The models with biologically inspired features perform better when we compare them against our PosScan baseline model.** MSE and ScanMatch results model comparison. A lower MSE represents a better prediction of the models, where 0 is it is lower bound. ScanMatch is measured between 0 and 1 when the latter indicates a better prediction.

[7]"G" from global and "L" from local.

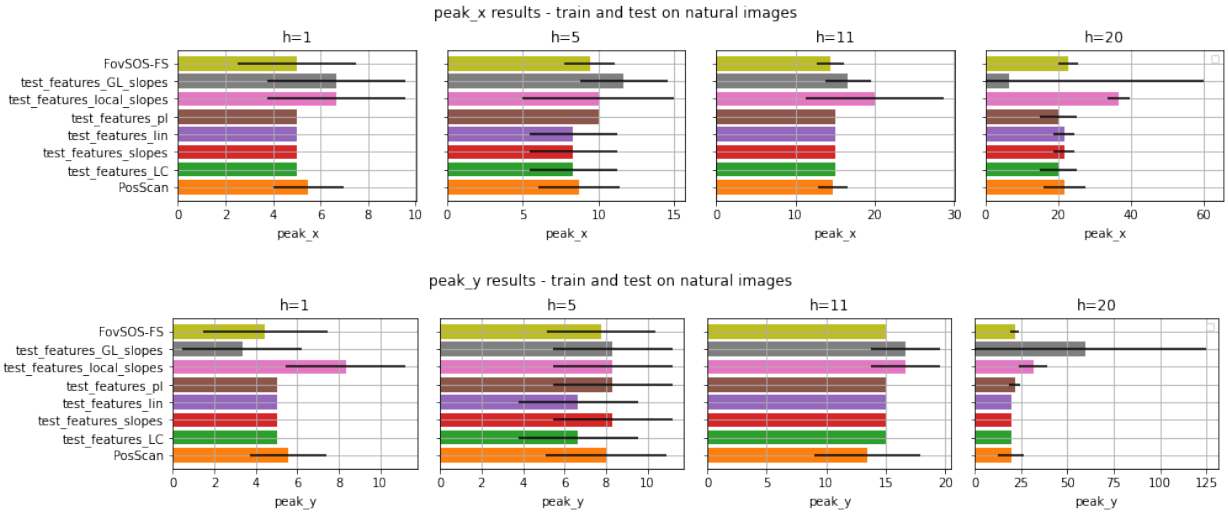

Figure 4.22: **The models with biologically inspired features perform better when we compare them against our PosScan baseline model.** Cross-correlogram peaks model comparison. Positive values indicate the prediction lag is after the ground truth and negative values, and negative values indicate that the prediction is before the ground truth.

When we add the GL slopes and the local slopes features, the models get better on MSE (amplitude), but worse on predicting the temporal phase of scanpaths. The latter results can be seen on the ScanMatch, MM Duration, and cross-correlogram peaks where both models are the worst compared with the others. About spatial results in general there is no improvement with respect to others when adding these features (see spatial MultiMatch metrics). We speculate that the local features are those that cause these poor results, not because local features are uninformative for predicting but because of the hyper-parameters used to extract the local features (like the selected boundary or model adjustments). We decided not to delve further into tuning model hyper-parameters because all the models have the same architecture, except for FovSOS-FS architecture which will be explained later (for now just think of it as a scanpath predictor).

The results above suggest that spatial information plays an important role as an input feature for models to predict where people will see. Then, as we discuss in Section 2.5 subjects focus their attention without the same importance based on where they looked or what they considered relevant to look, this could be thought of as a sequential selection of which zone is important. We believe that this is because when people are free exploring a new scene, they focus their attention on the areas which they consider most salient, and the tested features somehow encapsulate this spatial information. In this sense, we started to think about how to add this information to models, but this spatial information must be dependent on where the subject is seeing i.e. it must be mutable on time and dependent on it.

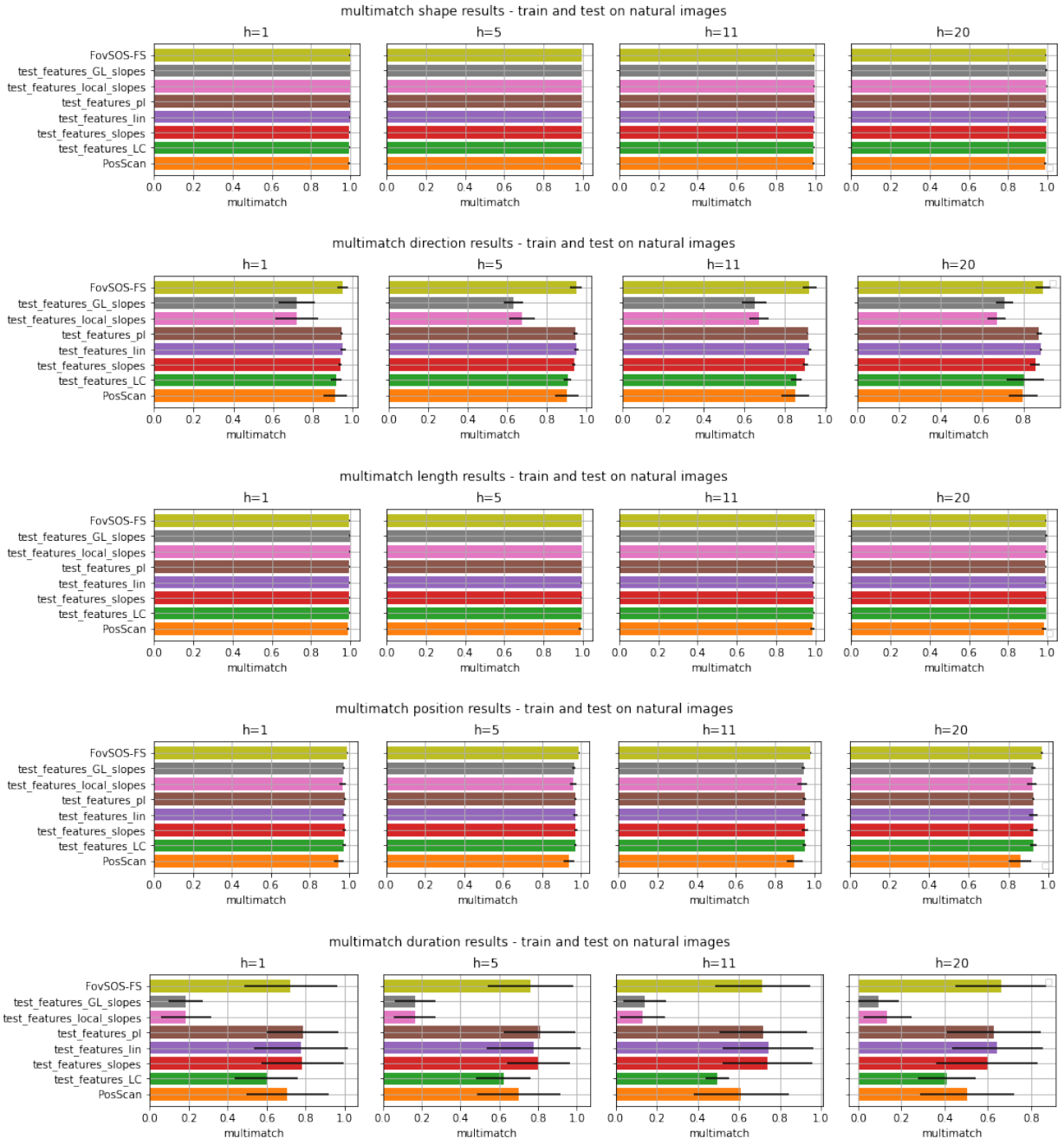

Figure 4.23: **The models with biologically inspired features perform better when we compare them against our PosScan baseline model.** Multimatch model comparison. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

We continue our search for features that could improve our first baseline model. Literature suggests that foveated images that simulate the subjects' fovea centered on where they are seeing (see Section 2.7 for more details) could improve our results. In addition, note that this feature has the requirements that we mentioned above i.e. it has spatial information embedded in the foveated image and it is mutable on time for the center of the fovea on where subjects are seeing. However, it is not directly adding this information to our model, so we had to think about how

to compress the image content for adding this feature to our model. Our first approach was to use a pre-trained CNN called Mobilenetv2 [Sandler et al., 2018] and extract from it the last dense layer as a new feature for our PosScan model. Unfortunately, models slightly improve when we measure the MSE loss on the validation set[8]. Anyway, we believe that this approach could work, but further exploration of hyper-parameters or use a fine-tuned version of some CNN is needed, finally we prefer to leave this CNNs exploration for extracting features from foveated images as future work, but we continue testing with foveated images.

We test models changing their architecture by modifying components like the number of layers, units on every layer, dropout rates, optimizer, learning rate, loss, and its core layer LSTM. We found that using attention layers as the core of our model benefits the optimization phase, and so helps to find better models for scanpath prediction. This issue will be discussed in detail in the next section with our novel model "Foveated Saliency and Ocular Scanpath with Feature Selection" (FovSOS-FS).

4.4 Modelling scanpath with an attention neural model using positional and spatial information though time

In the previous section, we saw most of our thoughts that led us to think that our architecture must include spatial information through time embedded in it. In this sense, we also realized how important is the positional information for predicting scanpaths in our PosScan model. Hereby, we create our model based on the above assumptions by keeping the position information as input and adding variable saliency maps depending on foveated images calculated through time. This model is called "Foveated Saliency and Ocular Scanpath with Feature Selection" (FovSOS-FS).

4.4.1 Saliency maps from foveated images and Attention model

The proposed FovSOS-FS model is a variant of the PosScan model which has been enhanced in order to take into account the ocular scanpath and saliency maps (variable on time). The Ocular scanpath information is used as input in the same way as it was explained for our PosScan model. For saliency maps instead, we had to do some feature engineering.

Saliency has the spatial information of where subjects saw embedded, but to meet that it has to be mutable and dependent on time, we opt to calculate the saliency of foveated images. With this, the saliency map will vary on time because the foveated images are dependent on the zone where the subject is seeing. To ensure the mutability of saliency maps we choose a convolutional neural model which calculates saliency called SALICON [Huang et al., 2015] (for more details of this architecture see Section 2.7). It is important to note that outputs obtained from CNNs change even with small variations in input such as fove the input image in different zones. In Figure 4.24 we can see how the saliency maps obtained from foveated images vary when the fovea is centered in different zones, the first column shows the image and its respective saliency map without any foveation, in the second and third columns we can see how the foveated saliency map change it attention (saliency) from lower left zone to the upper right zone of the foveated image, respectively.

Our tests reveal that is easier for the model to find the optimum (minimizing the MSE) when we

[8]Train and test were performed only in natural images.

take just some pixels (the important ones) of the saliency images and not by using the whole map. For that, we employ a feature selection module which uses Extra-Trees Regressor [Geurts et al., 2006] that fits several randomized decision trees on various sub-samples of the training dataset, with this we can select the most important pixels for the model by using the Gini importance [Menze et al., 2009]. Followed by this feature selection, we decompose the features using Principal Component Analysis (PCA) [Wold et al., 1987] which allow us to reduce the dimensionality by selecting the number of components based on the amount of variance explained by each of the selected components, we choose that percentage to be greater than 95%.

After the feature selection, we concatenate the ocular scanpath features with the selected pixels from saliency maps. With these features, we feed our model to predict the next position (one step ahead) where the subject might look.

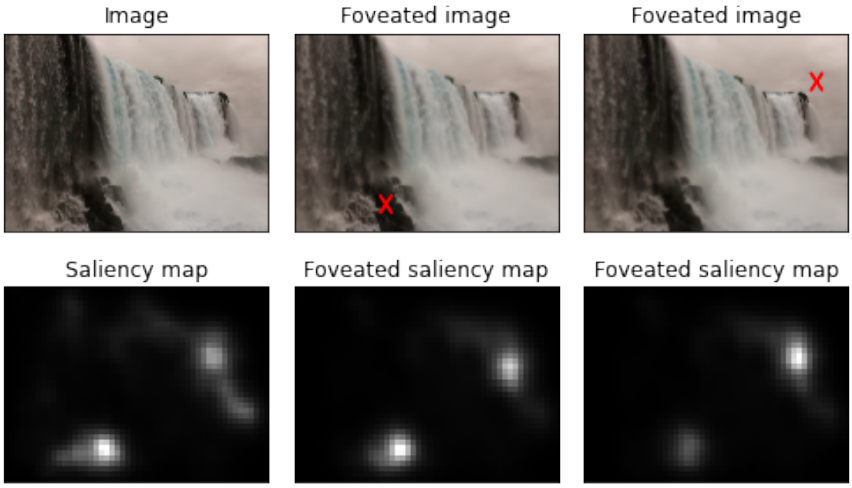

Figure 4.24: Calculation of foveated images and their respective saliency maps. The red cross represents where the foveation is centered (the cross is just for illustrative purposes).

We made many architecture tuning tests which gave us the intuition that Attention layers[9] [Vaswani et al., 2017] works better than LSTM layers in our task. In fact, Attention has been progressively replacing the RNN models (like LSTM) since Attention provides the context of any position in the input sequence by calculating the attention weights between every input (token) simultaneously, unlike RNNs which sequentially process the data so when it has to work with longer sequences it starts to forget the context of the data from the beginning of the sequence. Another advantage of the Attention layer is its parallelism and speed to be trained because it only performs dot-product operations for the entire sequence simultaneously. Parallelism has played an important role in recent years in machine learning research and thanks to the GPU and TPU processing multiple and simultaneous computations have been made possible.

In Figure 4.25, the whole FovSOS-FS model is shown. Note that we opt to create an attention-based module for making predictions. This Attention module is almost the same as the LSTM module in our PosScan model, but here we replace the LSTM layer with a Multi-Head Attention

[9]Attention layers are the building blocks of Transformers, which are state-of-the-art in NLP models.

layer and a Global Average Pooling layer. As hyper-parameters of our model, we use: 5 number of attention heads, 5 is the size of each attention head for query and key, and 0.4 of dropout for the Multi-Head Attention layer; Dropout layers has a rate of 0.3 to drop; Feed Forward layers has 80 and 100 units with ReLU activation respectively. We minimize the MSE loss, we also test with other losses but MSE achieves the best performances for scanpath prediction.

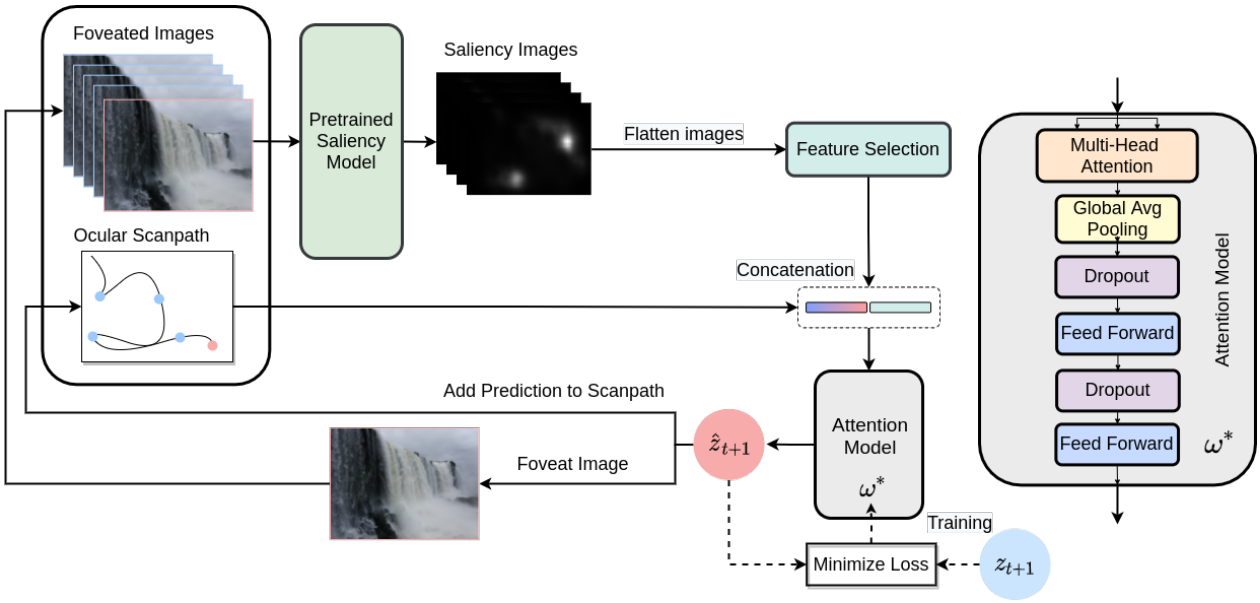

Figure 4.25: **FovSOS-FS architecture**. Prediction 1-step ahead with sequences of length $\ell = 5$. Note that the foveation in the image is centered in its corresponding stare position. Also, note that Ocular Scanpath is a tensor with the stare positions through time. Blue circles represent the available data at time t and the red one is the prediction.

4.4.2 Analysis of FovSOS-FS model

For our analysis we did two tests, for the first we train our model using data from natural images and predicting with a recursive forecast. Since the predictions take a lot of computational time with this forecasting strategy, for the second test we decided to change this to a direct forecast.

We want to emphasize that the recursive forecast for the FovSOS-FS model supposes to create a foveated image from every prediction, this process imposes a computational time constraint to compute the steps-ahead predictions. For this, we opt to train our model using only scanpaths retrieved when people free-view natural images. Recall that the PosScan results indicate that when grouping the results by training image, images with high visual content are the best for predicting scan paths, so natural images emerge as our first choice of interest.

Considering the above, it becomes very expensive to get samples and calculate the uncertainty via MC-Dropout, for this in Figure 4.26 we report only the scanpath prediction without the uncertainty. We can see that as the horizon steps ahead increase, the prediction amplitude gets worse. Also, the time-shift between the ground truth and the prediction increases i.e. the model takes longer to predict when the saccades will be performed. We believe that the above could be due to the propagated error since we are using recursive forecast, so in a later section we change this forecasting strategy to direct forecasting.

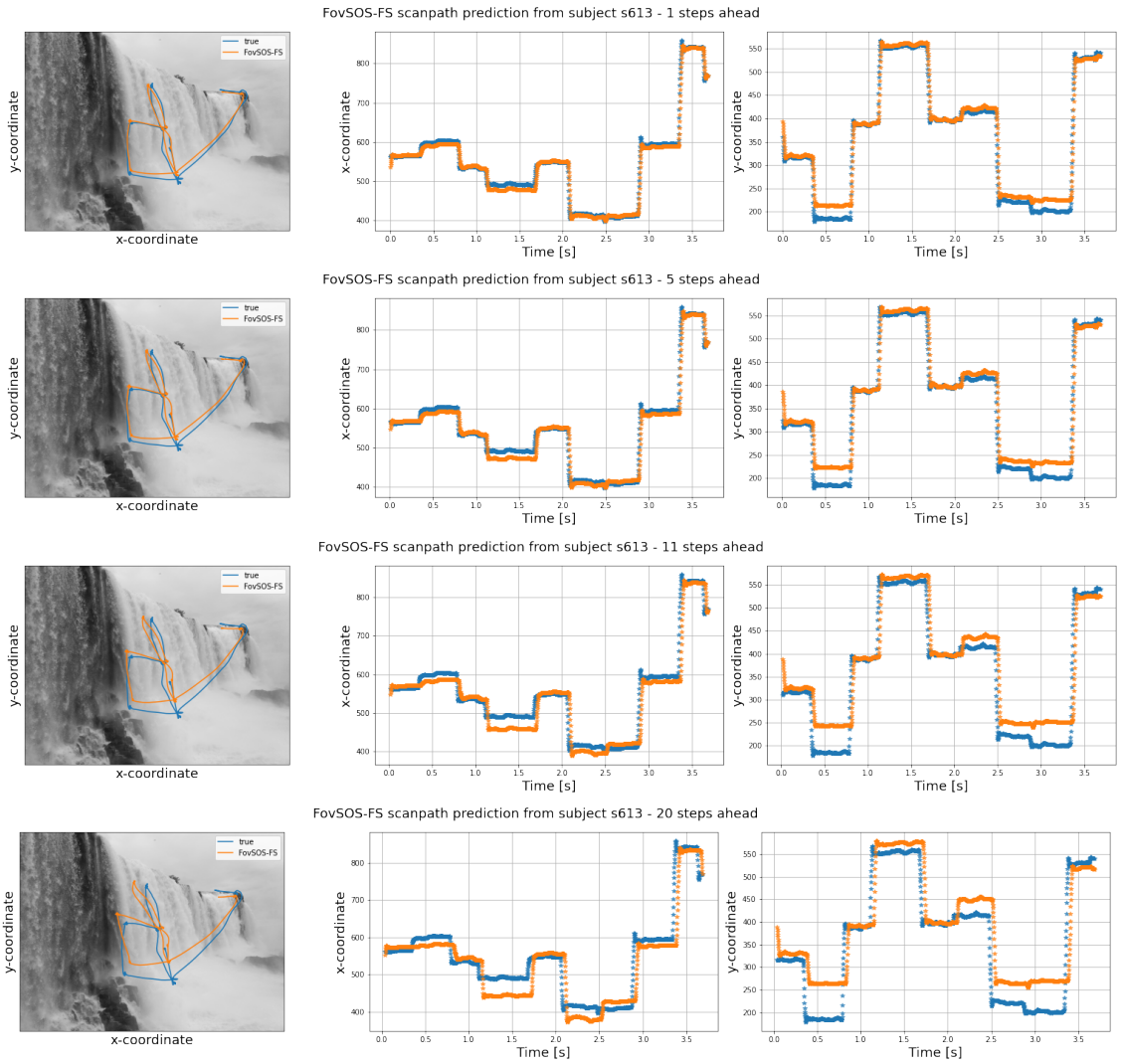

Figure 4.26: **The prediction gets worse as we increase the steps-ahead horizon since the error is propagated through the steps ahead prediction.** Scanpath prediction from subject s613 when free-viewing a natural image.

4.4.2.1 FovSOS-FS results grouped by predicted image type

To analyze how well the FovSOS-FS model can predict scanpaths when grouped by image type, we first looked at the MSE error and the ScanMatch for the different image types. The results are shown in Figure 4.27. From the Figure, we see that the MSE is lower for the groups that have higher visual content. This makes sense, as we would expect information from images with more visual content to be captured more when using our salient foveated images. Surprisingly, unlike the other images with high visual content, prediction in pink noise images start to decrease in performance when the prediction horizon increases. On the other hand, images with low visual content have a higher MSE error. This is due to the fact that there is not much information to be gleaned from these types of images in the features obtained from the salient foveated images, they are all constant values so the network does not benefit from this new concatenated feature. The ScanMatch shows a similar trend as the MSE error, where the natural images predictions seem to be the better predicted.

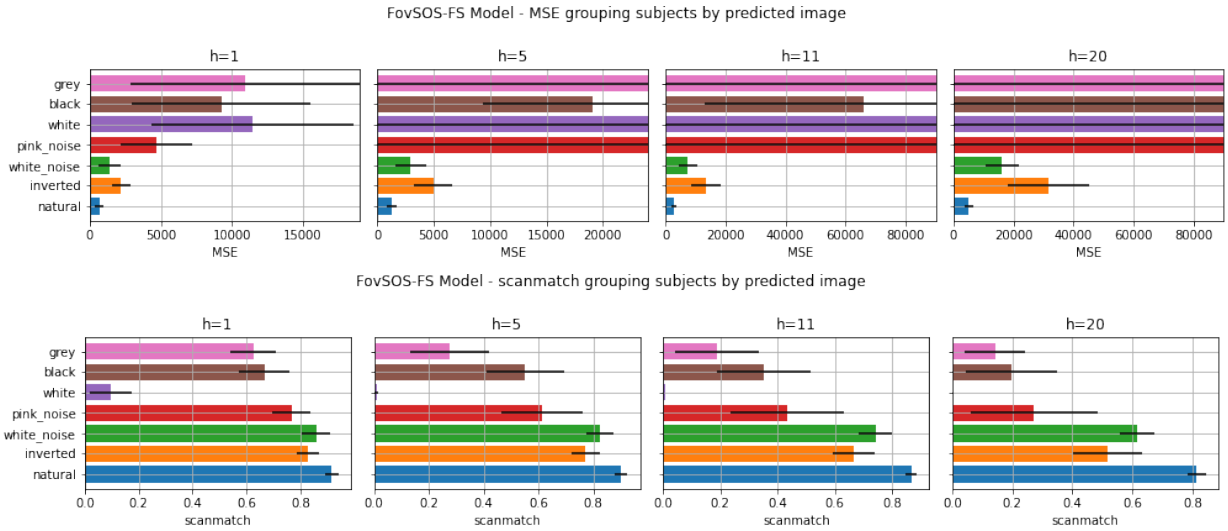

Figure 4.27: **The MSE and ScanMatch show a similar trend where the higher visual content image type predictions are better predicted, this is because the information from images this image types are better captured since we added the salient foveated images as features.** FovSOS-FS metrics results measured with MSE and ScanMatch, image types grouped by predicted image. These metrics allow us to compare spatial differences between the predictions and the scanpath ground truth. Note that ScanMatch also measures the temporal phase error between scanpaths. A lower MSE represents a better prediction of the models, where 0 is it is lower bound. ScanMatch is measured between 0 and 1 when the latter indicates a better prediction.

With respect to other metrics, in Figure 4.28 the MultiMatch metrics distribution shows a similar trend as the MSE and ScanMatch, where the group of images with high visual content has better predictions (with the pink noise exception). In Figure 4.29 we report the peaks from the cross-correlogram, where for the higher visual content group results are similar than the reported for our PosScan model, but for the lower visual content group results worsen.

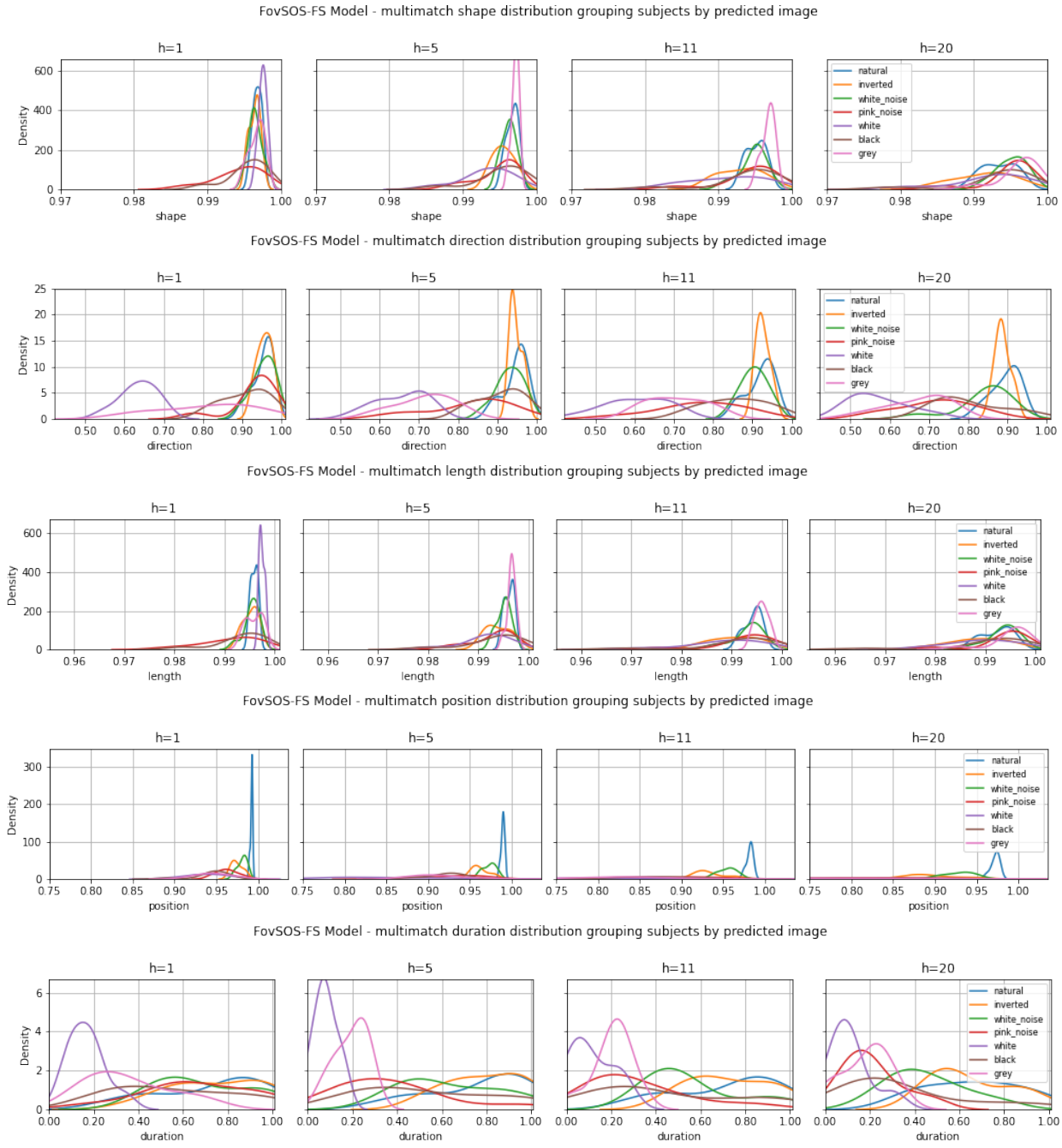

Figure 4.28: **The MultiMatch metrics distribution shows a similar trend as the MSE and ScanMatch where the group of images with higher visual contents has better predictions.** FovSOS-FS distribution results using MultiMatch metrics, image types grouped by predicted image. Note that the first four (rows) MM metrics are in charge of measuring spatial features between the scanpaths (predicted and ground truth), and the last MM metric duration is the only one that measures temporal errors. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

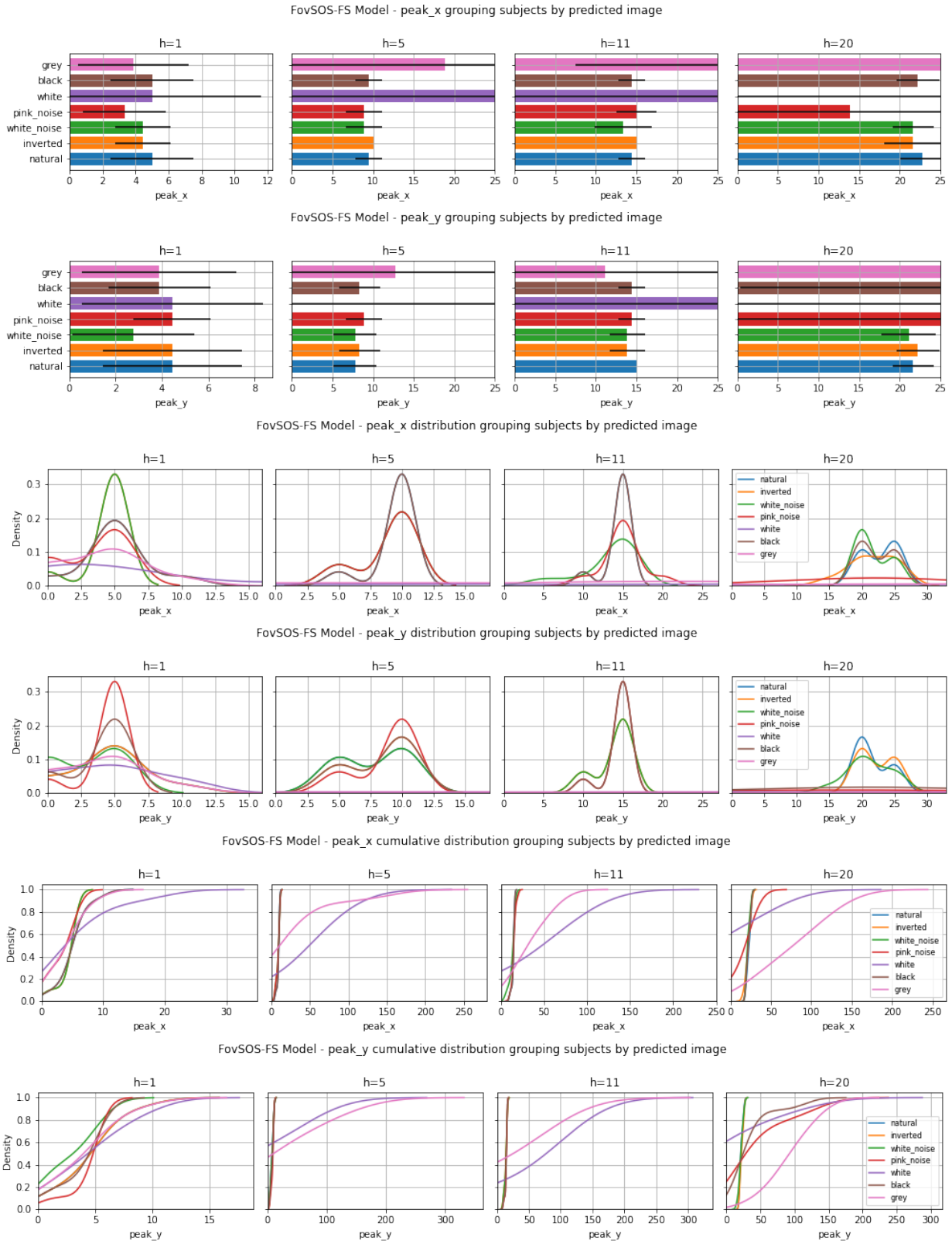

Figure 4.29: **The results in FovSOS-FS are similar to the reported in our PosScan model, but for the lower visual content group results worsen.** FovSOS-FS cross-correlogram peaks results grouped by predicted image. The first four rows are a histogram and the distribution of the cross-correlogram peaks, and the latter two rows are the cumulative distribution of the cross-correlogram peaks. Positive values indicate the prediction lag is after the ground truth and negative values, and negative values indicate that the prediction is before the ground truth.

4.4.3 Analysis of FovSOS-FSD

As we mentioned in the former sections, results suggest that as we increase the horizon steps ahead the error propagated by using a recursive forecast begins to be more significant. Sometimes even losing the ability to predict saccades and being able to just follow the scanpaths. While the uncertainty in fixations is greater, it will be more difficult for the model to predict when the saccade will take place. Since the direct forecast is computationally less expensive and we could pre-compute the saliency images features, we were able to train and predict for each type of image. For more clarity, FovSOS-FSD is our FovSOS-FS model but with the direct forecast strategy.

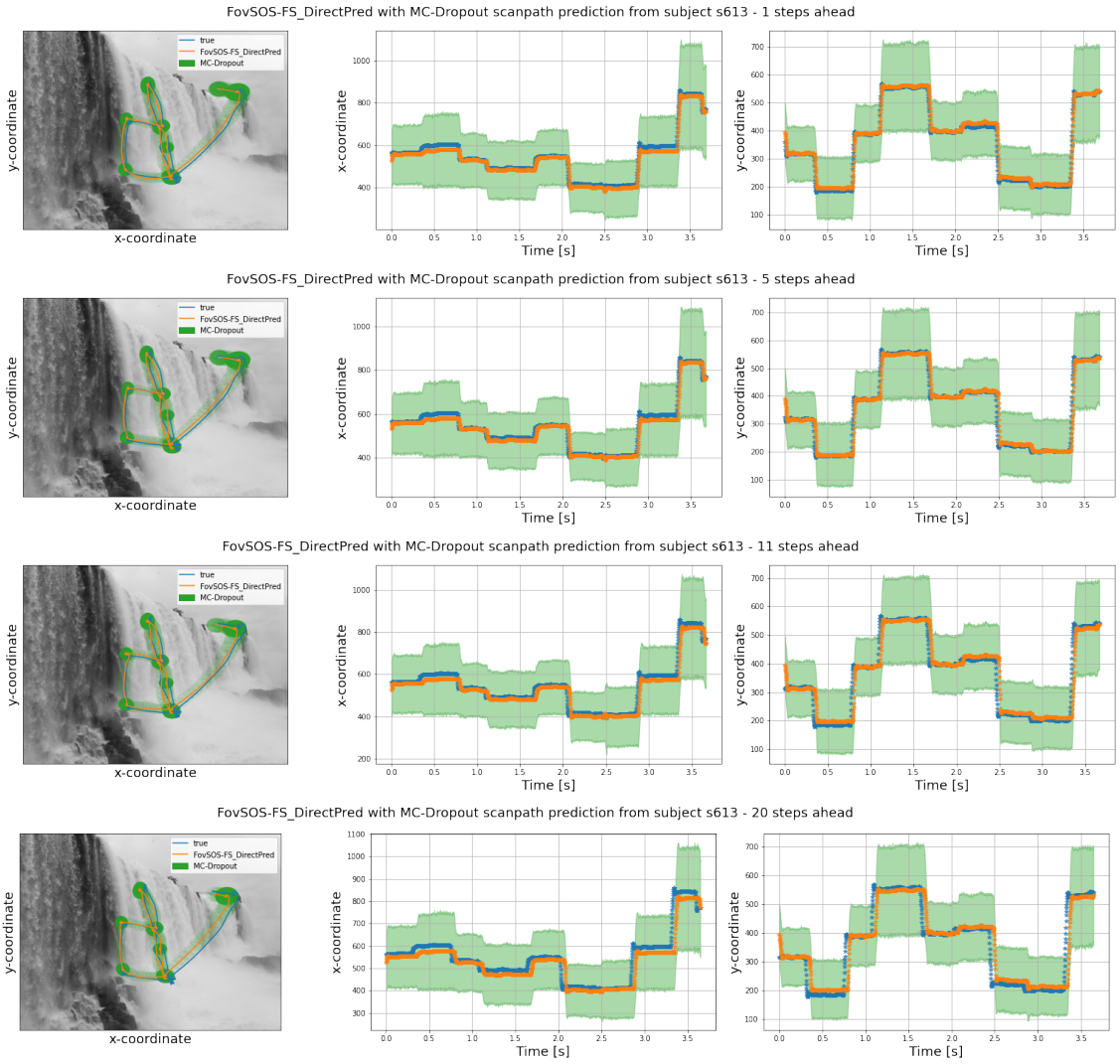

Figure 4.30: **Our FovSOS-FSD model can predict scanpaths even at higher horizons thanks to the information of the foveated saliency maps and the ocular scanpath.** Scanpath prediction from subject s613 when free-viewing a natural image.

In Figure 4.30, we reported the scanpath predictions (from subject s613) with MC-Dropout of our FovSOS-FSD model, in that we can see how our FovSOS-FSD model can predict scanpaths well even at higher horizons. This fact is not trivial because direct forecasting was attempted with our previous PosScan model, but we did not obtain the expected results. That FovSOS-FSD can be

fitted for higher horizons is thanks to the information of the foveated saliency maps (and the feature selection module) that allows the model finds and optimum for higher steps ahead. In addition, note how the uncertainty decreases compared with our PosScan model, and the differences between the uncertainty in fixations and saccades are more noticeable, in other words, it is easier for the model to know where the saccade will land once a saccade has already started (less uncertainty in saccades), but it is more difficult to predict when the saccade will start while the subject is fixating.

4.4.3.1 FovSOS-FSD results grouped by train image

We group results by train image type, with this we can compare the general performance with our previous PosScan model. The results are shown in Figure 4.31, we can see that results look alike with the PosScan model since the grouping by visual content image types i.e. higher visual content image types perform better than lower visual content image types. In addition, we found an improvement in the prediction when using direct prediction instead of the recurrent forecast. That is although we increase the prediction horizon the results remain almost without changes (similar error values), in other words, the model just worsens slightly when predicting more steps forward. This is because for each step forward we trained a different model, which allows to them find their optimum, and so models could correctly forecast independently of the predicted step ahead. The above applies for MSE and ScanMatch, if we compare them with PosScan results in Figure 4.5, we see how the MSE increases (or ScanMatch decreases) with the prediction horizon.

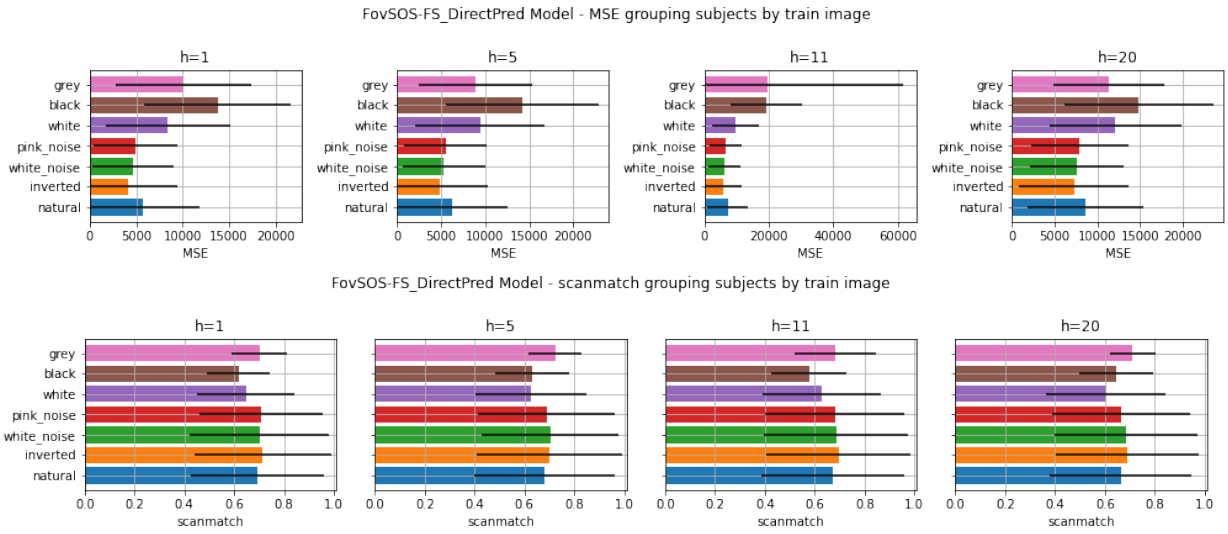

Figure 4.31: **We found an improvement in the prediction when using direct prediction, as we increase the prediction horizon the results do not worsen as in the our previous models.** FovSOS-FSD metrics results measured with MSE and ScanMatch, image types grouped by train image. These metrics allow us to compare spatial differences between the predictions and the scanpath ground truth. Note that ScanMatch also measures the temporal phase error between scanpaths.

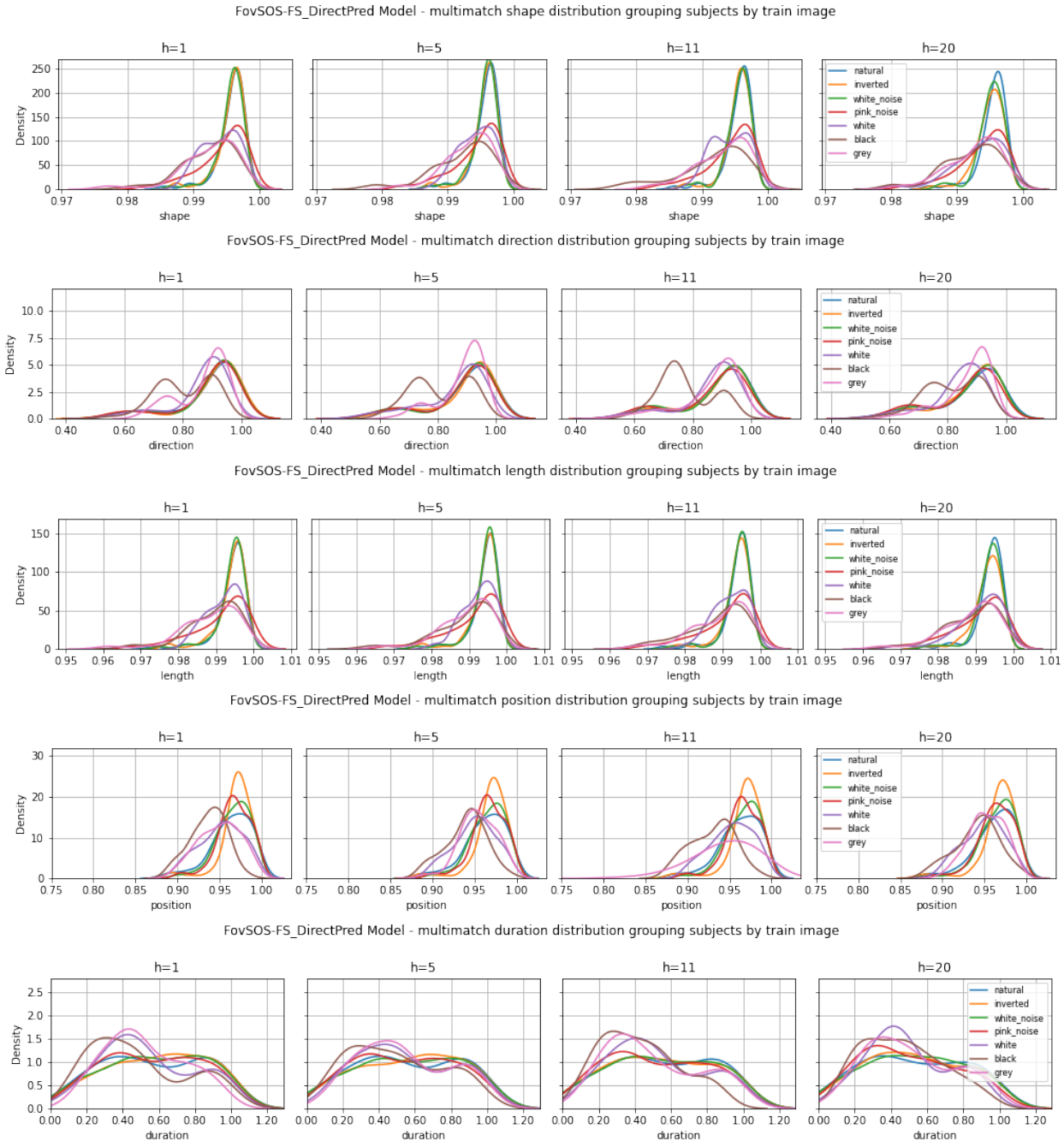

Figure 4.32: **FovSOS-FSD performs better than the PosScan and FovSOS-FS when increasing the prediction horizon.** FovSOS-FS direct prediction distribution results using MultiMatch metrics, image types grouped by predicted image. Note that the first four (rows) MM metrics are in charge of measuring spatial features between the scanpaths (predicted and ground truth), and the last MM metric duration is the only one that measures temporal errors. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

When we compare Multimatch results in Figure 4.32 with the PosScan results in Figure 4.7 we see two cases:

- The first case is an improvement of our FovSOS-FSD with respect to the PosScan and

FovSOS-FS when increasing the prediction horizon. This can be seen in MM shape, MM length, and MM position.

- The second case the visual content image type makes the difference. High visual content image types perform well in both models, but for lower visual content images results are worse in the FovSOS-FSD model than the PosScan model. This assumption is true for low prediction horizons because when we increase the horizon FovSOS-FSD still achieves good performance and PosScan begins to deteriorate its predictions. This can be seen in MM direction and MM duration.

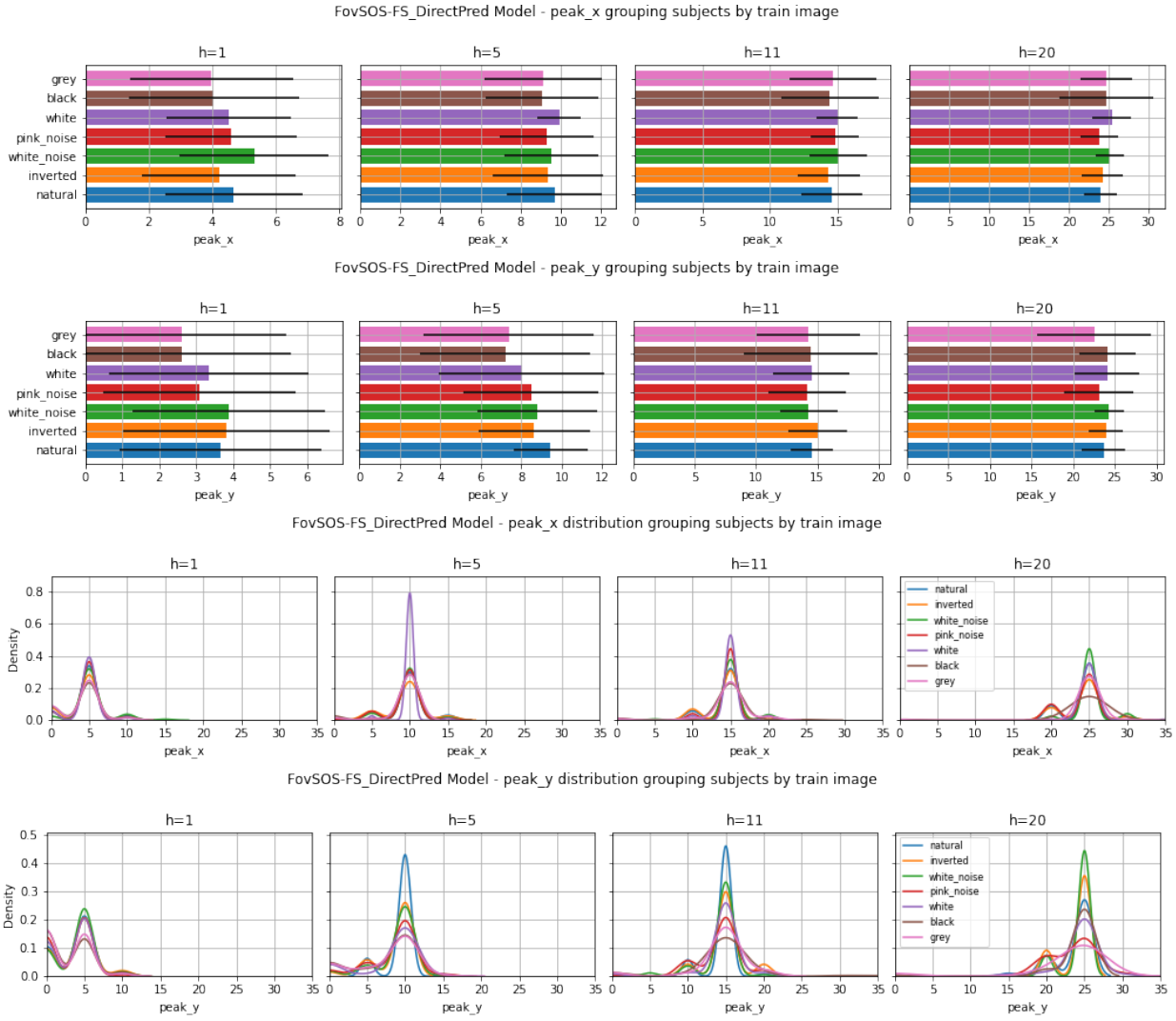

Figure 4.33: **Slightly improvements in the distribution of the cross-correlogram peaks with respect to our previous models.** FovSOS-FS cross-correlogram peaks results grouped by trained image. The first two rows are the cross-correlogram peaks and the latter two rows are the distribution of the cross-correlogram peaks. Positive values indicate the prediction lag is after the ground truth and negative values, and negative values indicate that the prediction is before the ground truth.

With respect to the peaks of the cross-correlogram in Figure 4.33, we can see an improvement only on the y position (height in the image) for the one-step prediction ahead when we compare

it with the PosScan results in Figure 4.9. For others rather than the above case, we cannot see enough differences between models. This could be since both models do not have in their loss function a term that involves time lag, and so neither of them manages to improve this metric at least via MSE loss minimization.

4.4.3.2 FovSOS-FSD results grouped by predicted image

We group results by predicted image type, with this we can compare the general performance of the FovSOS-FSD model with respect to our previous PosScan and FovSOS-FS models.

In Figure 4.34 we can see the results from FovSOS-FSD when grouping by predicted image type. When we compare these results with the PosScan model in Figure 4.12 we see how FovSOS-FSD performs better for every metric and image type with the exception of the white image type. On the other hand, compared with the FovSOS-FS model, the results of its direct forecast version (FovSOS-FSD) are more stable or, in other words, FovSOS-FS has a huge difference in performance between the higher and lower visual content image types. About this last statement, it is up to the user to choose which model to use depending on the type of image that they want to predict and the acceptable confidence that the operator should need, at least for spatial errors measured by the MSE and the ScanMatch because this does not imply a generalization for temporal errors that models could have.

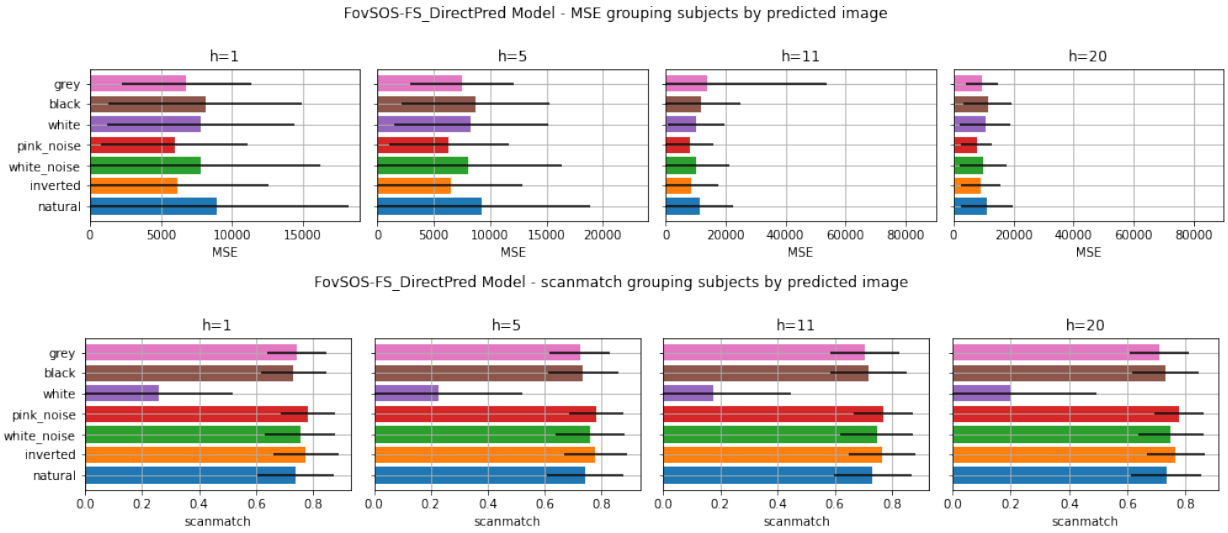

Figure 4.34: **Compared with our previous models FovSOS-FSD performs better on MSE and ScanMatch metrics with the exception of the white image type.** FovSOS-FSD metrics results measured with MSE and ScanMatch, image types grouped by predicted image. These metrics allow us to compare spatial differences between the predictions and the scanpath ground truth. Note that ScanMatch also measures the temporal phase error between scanpaths.

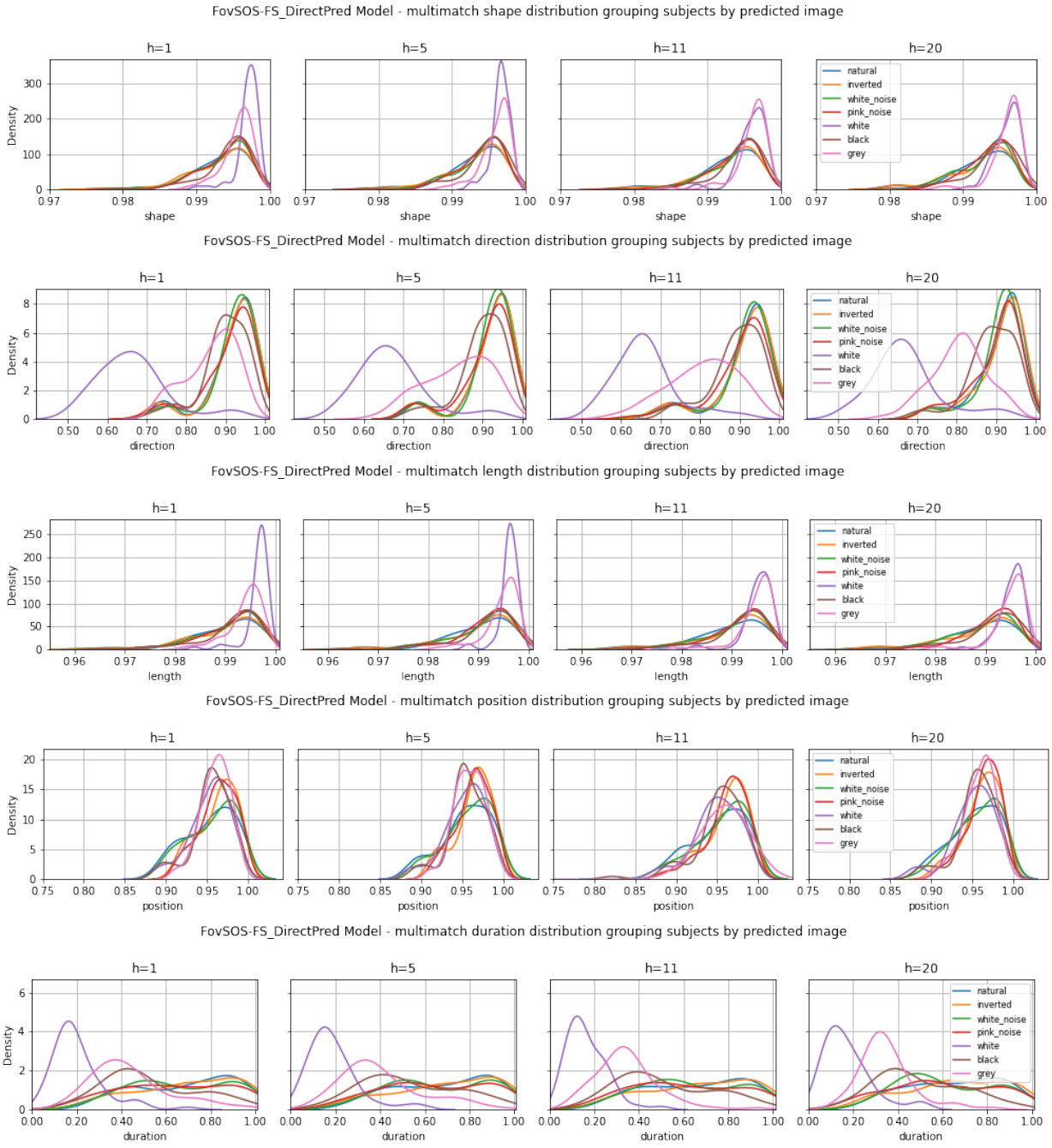

Figure 4.35: **FovSOS-FSD outperforms PosScan and FovSOS-FSD in general, where the major differences appear when increasing the prediction horizon.** FovSOS-FS direct prediction distribution results using MultiMatch metrics, image types grouped by predicted image. Note that the first four (rows) MM metrics are in charge of measuring spatial features between the scanpaths (predicted and ground truth), and the last MM metric duration is the only one that measures temporal errors. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

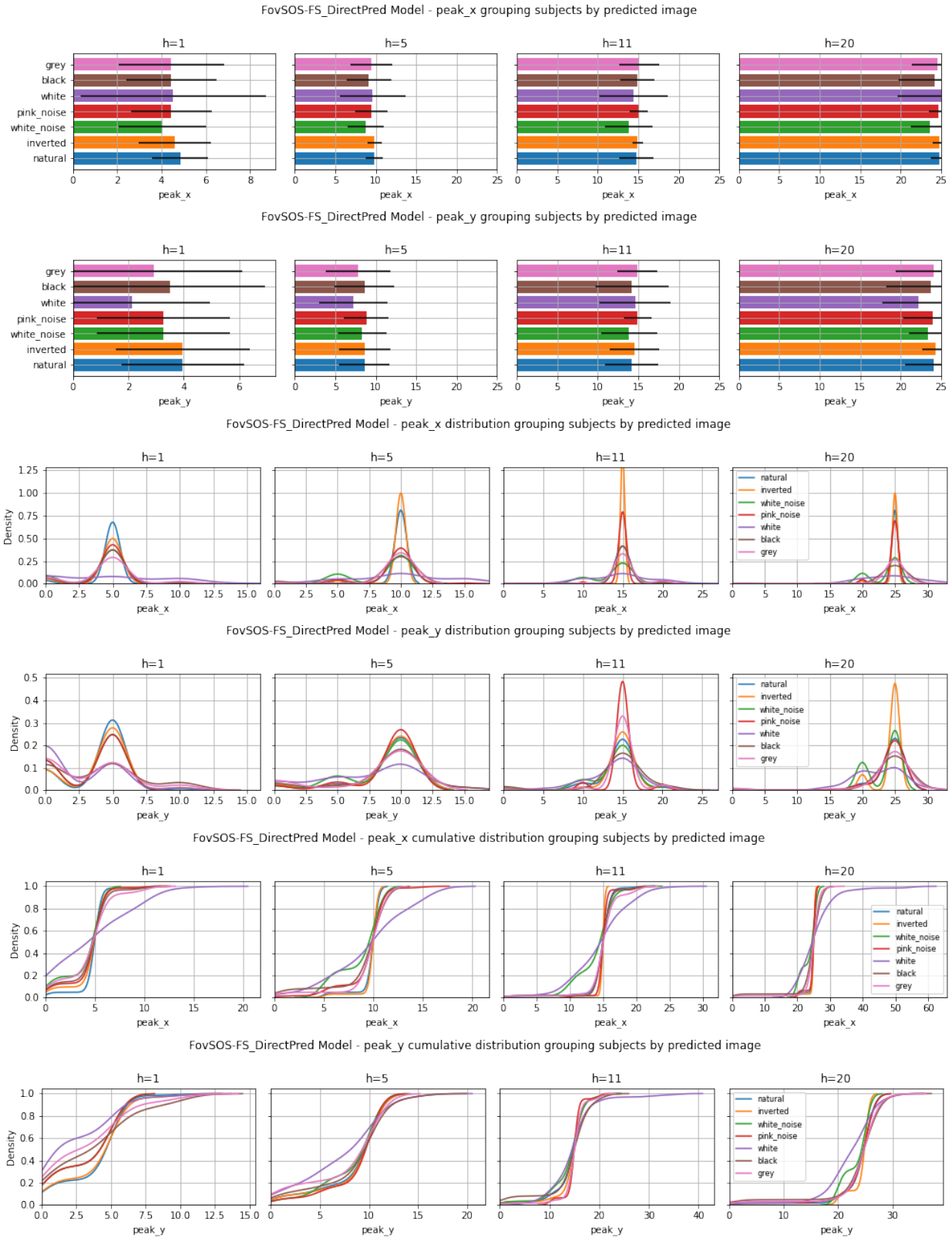

Figure 4.36: **The cross-correlogram peaks distribution does not show major differences between FovSOS-FSD and our previous models (PosScan and FovSOS-FS). FovSOS-FS cross-correlogram peaks results grouped by predicted image.** The first two rows are the distribution of the cross-correlogram peaks and the latter two rows are the cumulative distribution of the cross-correlogram peaks. Positive values indicate the prediction lag is after the ground truth and negative values, and negative values indicate that the prediction is before the ground truth.

Multimatch metrics are displayed in Figure 4.35, compared with our PosScan and FovSOS-FS models in Figure 4.14 we can see that FovSOS-FSD outperforms PosScan in general, except in a few cases like MM direction and MM duration where for lower visual content image types FovSOS-FSD had worse performance. It is important to note, that the major differences in metrics between the two models are when we increase the prediction horizon, where we can see how the direct forecast helps the model keeps a good performance at higher prediction horizons. Following, when we compare FovSOS-FSD with FovSOS-FS MultiMatch results in Figure 4.28, when the latter predicts better image types with higher visual content than FovSOS-FSD even at higher prediction horizons.

About the cross-correlogram peaks in Figure 4.36 which represents the time lag between prediction and ground truth, we do not appreciate major differences between FovSOS-FSD and our previous models PosScan and FovSOS-FS. Only FovSOS-FS has a bad performance in images with low visual content as we have already mentioned in previous sections.

4.4.4 Remarks

We want to emphasize that these saliency maps obtained from the foveated images are important and help the model to find the optimum, which is not trivial, but they do not provide sufficient temporal information to predict a saccade beforehand the subject starts to perform it. In other words, the saliency obtained from the foveated images through time helps the model to predict better the spatial information but not the temporal information. For instance, it is necessary to still investigate how to reduce the time lag between predictions and ground truth scanpath.

The comparison between our PosScan and FovSOS-FS models show us how the saliency obtained from the foveated images helps the model to capture the context of the images, which is not possible with just the position of where the subject is looking. In this sense, we can say that our FovSOS-FS models capture the scanpaths information better than our PosScan model since the latter does not take into account the spatial information through time.

When we compare the recursive forecast with the direct forecast, we can see that the latter is computationally less expensive and is more capable of finding the optimum of the models, so model performance is better. While the direct forecast propagates the uncertainty whilst we increase the prediction horizon.

4.5 Comparative discussion of models

In this section, our objective is to summarize the comparative results between the models and to identify the factors that contribute to the better performance of the different models.

In Figure 4.37 scanpath predictions by the three models are displayed, we report a sampled version of 3 seconds of the whole scanpath for every subject and one sampled natural image. For all cases, we can see how PosScan is outperformed by both FovSOS-FS and FovSOS-FSD models, where both are quite similar to the original scanpath but when we increase the horizon FovSOS-FSD achieves better results.

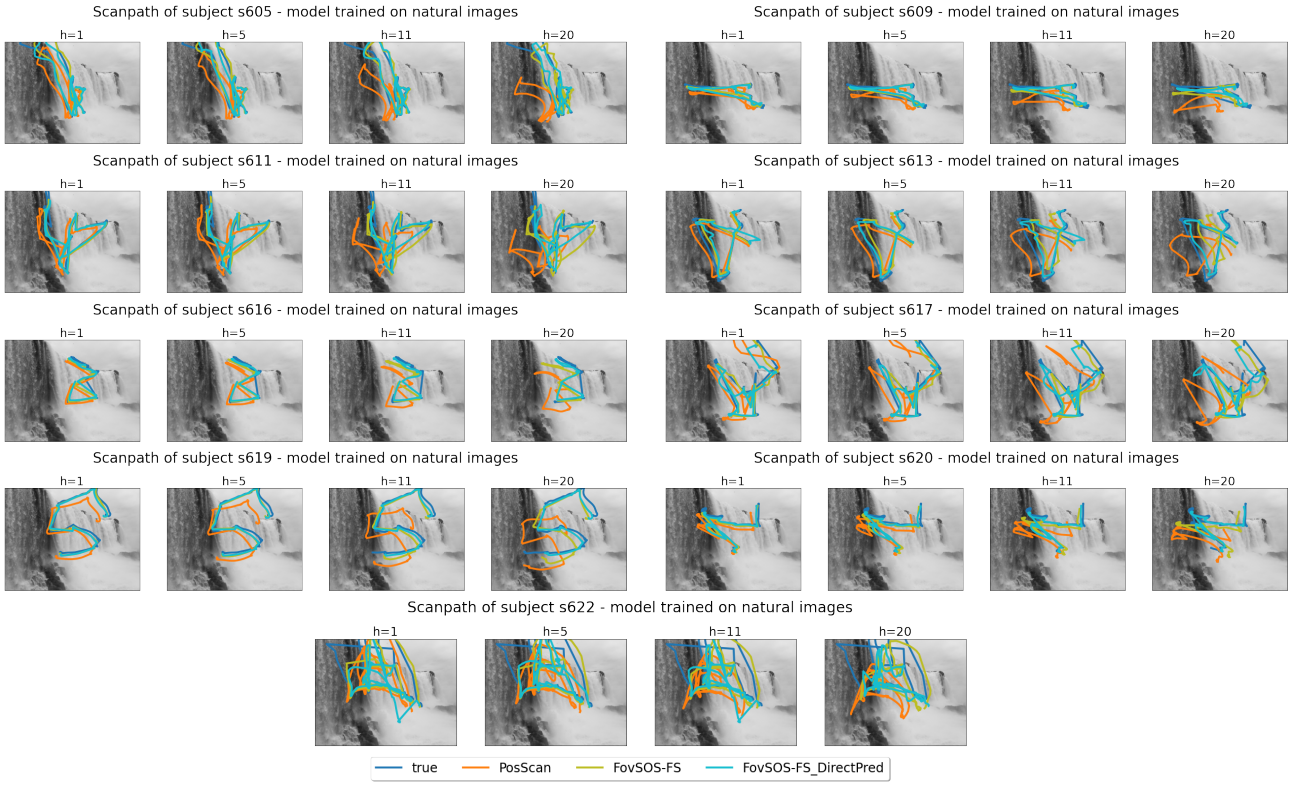

Figure 4.37: **PosScan is outperformed by both FovSOS-FS and FovSOS-FSD models, and when we increase the horizon FovSOS-FSD achieves the best results.** Sampled scanpaths prediction for every subject.

The main reason for the poor performance of PosScan in natural images compared to the others is that it does not take into account the information contained in the foveal region. FovSOS-FS and FovSOS-FSD models, by contrast, exploit the fact that the foveal region is where most of the information is concentrated and with this calculate a more consistent saliency map. This is particularly evident in the results for the natural images, where the models are able to predict the locations of the fixations with much higher accuracy than PosScan.

Another important factor that contributes to the better performance of the FovSOS-FS and FovSOS-FSD models is the use of attention layers. These layers allow the models to better capture the dynamics of the attentional process, as they can learn the importance of each data at different times without the forgetting problem that recurrent layers have.

It is worth noting that the FovSOS-FSD model achieves a better performance than the FovSOS-FS model at higher prediction horizons. This is thanks to the direct forecast strategy that allows the model to be trained and fitted specifically for a higher step ahead prediction. On the other hand, the recursive strategy inherently always passes an error to the next step, deteriorating its performance as the steps increase.

It is important to note that we also tested the direct forecast in our PosScan model, but the results were not good because the model could not find an optimum when minimizing the MSE loss at higher steps ahead (the model begins to overfit). We believe that the worse performance of PosScan when applying direct forecast is because the spatial information of the images is not

present in the data, so the model was not been able to extract more information only from the ocular scanpath that allows it to predict where subjects will look.

FovSOS-FSD took less time to calculate results than the FovSOS-FS model. This is because the FovSOS-FSD model just needed the saliency map for the current step which we pre-calculate just once. On the other hand, FovSOS-FS had to compute the saliency map at each step to re-feed the network (recursive forecasting strategy) which made this process computationally expensive.

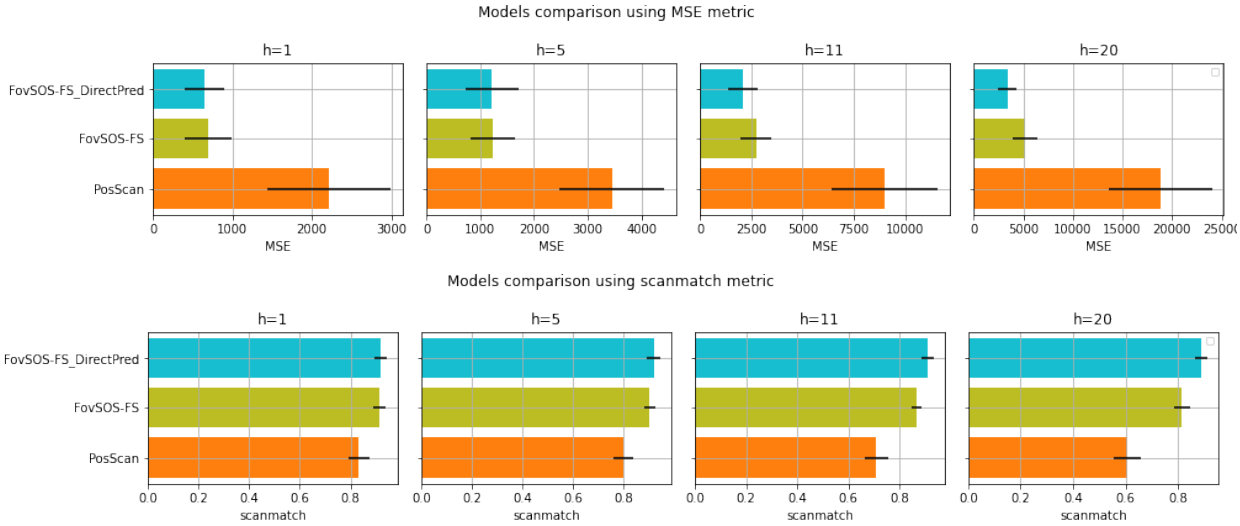

Figure 4.38: **FovSOS-FSD achieves a better performance than the FovSOS-FS model at higher steps ahead (11 and 20), and both models are better than PosScan.** Comparison of model predictions measured with MSE and ScanMatch. These metrics allow us to compare spatial differences between the predictions and the scanpath ground truth. Note that ScanMatch also measures the temporal phase error between scanpaths. A lower MSE represents a better prediction of the models, where 0 is it is lower bound. ScanMatch is measured between 0 and 1 when the latter indicates a better prediction.

We also want to emphasize the results metrics when training and predicting on natural images. In Figure 4.38 MSE and ScanMatch metrics are shown, we can see the same trend in these metrics as in the sampled scanpaths prediction results. FovSOS-FSD achieves a better performance than the FovSOS-FS model at higher steps ahead (11 and 20), both models are better than PosScan.

The above can also be seen in Figure 4.39, the same pattern is repeated when we measured with MultiMatch metrics. FovSOS-FSD achieves a better performance than the FovSOS-FS model at higher steps ahead (11 and 20), while PosScan is outperformed by both models.

The peaks of the cross-correlogram in Figure 4.40 show that at higher steps ahead performs almost similar i.e. the phase shift between the ground truth and the prediction are quite similar across all our models.

In general, we can conclude that the most complete prediction model is the FovSOS-FSD as it achieve the best results in terms of almost all the metrics. FovSOS-FS is also a very good model but when increasing the prediction horizon, FovSOS-FSD is much better.

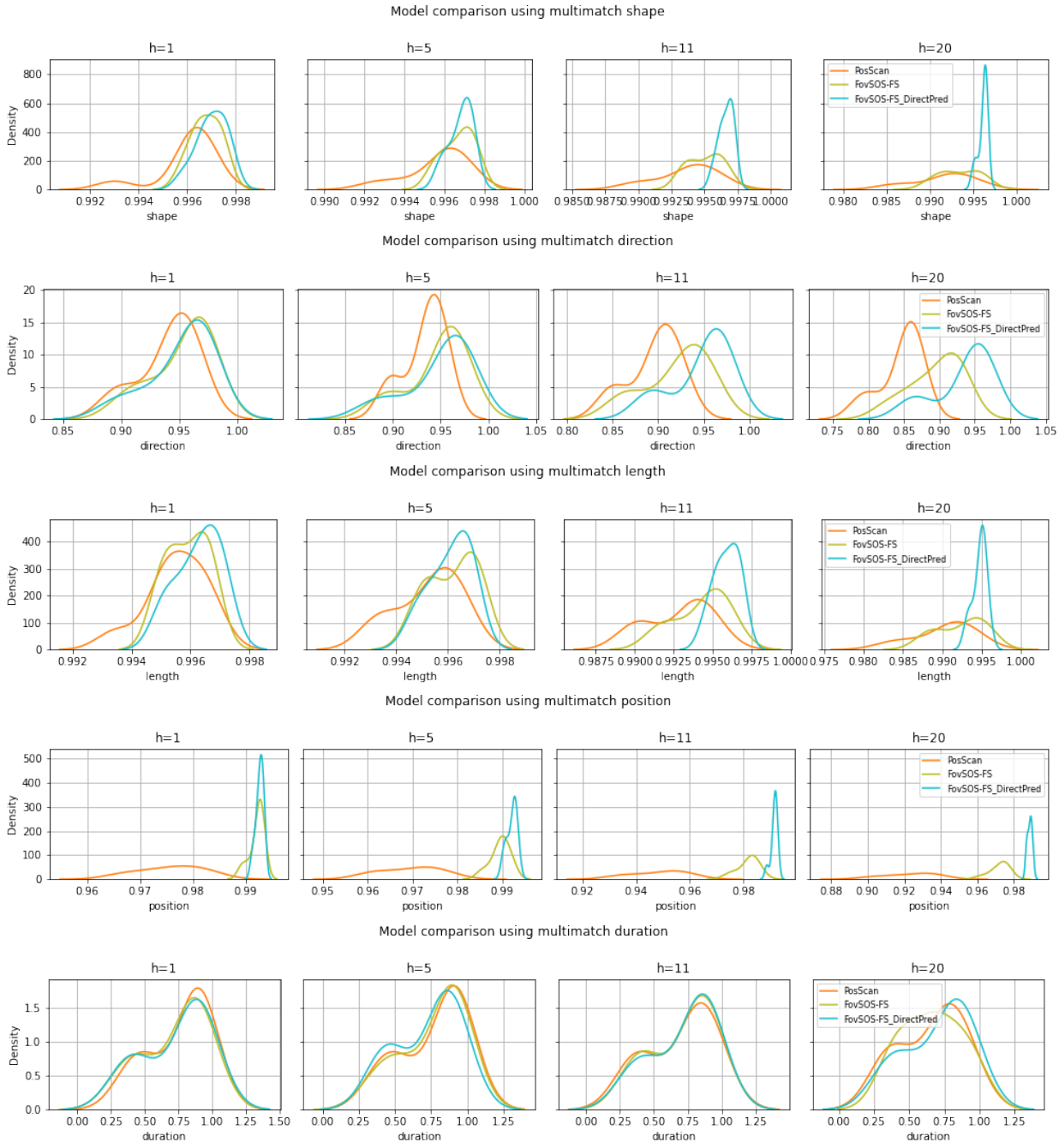

Figure 4.39: **FovSOS-FSD achieves better results than the FovSOS-FS model at higher steps ahead (11 and 20), while PosScan is outperformed by both models.** Comparison of model predictions measured with MultiMatch metrics. Note that the first four (rows) MM metrics are in charge of measuring spatial features between the scanpaths (predicted and ground truth), and the last MM metric duration is the only one that measures temporal errors. MultiMatch metrics are measured between 0 and 1 when the latter indicates a better prediction.

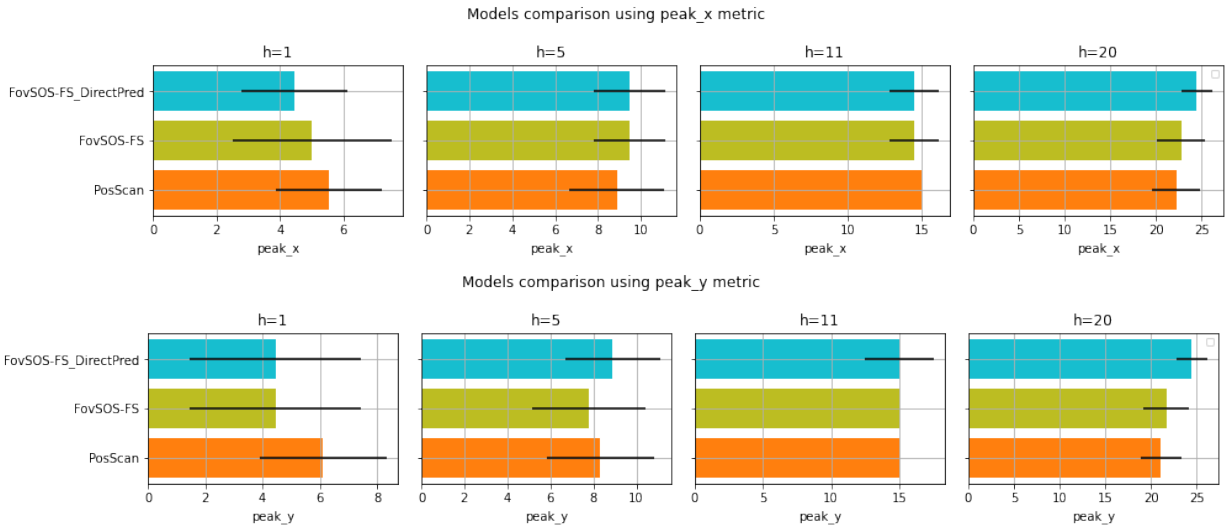

Figure 4.40: Comparison of model predictions measured with cross-correlogram peaks. Positive values indicate the prediction lag is after the ground truth and negative values, and negative values indicate that the prediction is before the ground truth.

Chapter 5

Conclusions and Future Work

In this work, we have proposed, implemented, and tested novel neural network architectures that infer the ocular scanpath of subjects when free-viewing images. We have also proposed a way to extract features from images (saliency maps from foveated images) to feed the model so that it can learn the visual content that subjects see, together with the previous subject scanpath the model can infer the next positions of subjects scanpath. Among our main findings is that saliency maps calculated from foveated images and the ocular scanpath of subjects have valuable information for making predictions, this is shown in the performance achieved by our FovSOS-FS and FovSOS-FSD models. Despite the above, we want to emphasize that these features do not provide enough temporal information to predict a saccade beforehand the subject starts to perform it. In other words, the saliency obtained from the foveated images through time helps the model to predict better the spatial information of scanpaths, but the temporal information obtained is not enough for the model to learn when a saccadic movement will be performed while the subject is fixating.

We analyze how well the models predict when they are trained with different image types, we find out that in general the models that were trained with higher visual content image types (high-frequency content) predicts scanpaths better, this could be since these kinds of images are more informative to the model for finding a better local-optima and make better predictions. In contrast, when training with lower visual content image types, the scanpaths in this kind of image type are more random when exploring because subjects have no saliency (to focus) in the scene, so they use only their default selection with no external influences, in other words, they are exploring without using their attentional mechanism. Because of this, if a model is trained with an image with lower visual contents, then it will hardly be able to predict when the subject is viewing images with higher visual contents since the model did not learn the subjects' attentional mechanism but only their default selection. The above behavior is congruent with literature, which suggests that during image free-exploration, the sensory-motor modulation over the early visual cortex acts when the explored scene has textures, but it is disengaged when an image is categorized as plain [Devia et al., 2017].

We also discuss how are the model predictions for different types of images no matter what type of image the model was trained on. In this sense, we realized that scanpaths obtained from subjects that free-view higher visual content image types are those in which it was more difficult to make a whole scanpath forecast. This is consistent with the mentioned above when grouping

results by train image type, since the models that were not been able to learn the attentional selection mechanism of subjects, such as the ones trained on scanpath from lower visual content image types, are not accurate. On the other hand, scanpaths obtained from subjects when free-view lower visual content image types are more easily to predict, especially when the model has already learned the attentional selection mechanism (when trained on scanpaths from higher visual content image types).

We can summarize the main findings in our predictive models as:

- The FovSOS-FSD model achieves the best results overall for scanpath prediction.
- The FovSOS-FS and FovSOS-FSD results are almost similar at lowers step ahead prediction, but at higher prediction horizons FovSOS-FSD model achieves a better performance than the FovSOS-FS model. This is thanks to the direct forecast strategy that allows the model to be trained and fitted specifically for a higher step ahead prediction.
- The main reason for the poor performance of PosScan in natural images compared to the others is that it does not take into account the visual information that subjects are seeing. FovSOS-FS and FovSOS-FSD models, by contrast, exploit the fact that the foveal region is where most of the information is concentrated and with this calculate a more consistent saliency map. Then, with this information as a feature, the model improves significantly its performance when predicting.
- The use of attention layers allows the models, FovSOS-FS and FovSOS-FSD, to better capture the dynamics of the attentional process, as they are able to learn the importance of each data at different times without the forgetting problem that recurrent layers have. In contrast, to the PosScan model that uses recurrent layers (LSTM) that have the vanishing problem when the sequences are too large.

We also explore the uncertainty differences between models by implementing the Monte Carlo Dropout, finding that the uncertainty decreases in our FovSOS-FSD model compared with our PosScan model. With this technique, it is possible to notice how the model uncertainty changes when adding new features, so it helps to select features that reduce the uncertainty. That FovSOS-FSD has just a low uncertainty in the fixations reveals that although the model has improved a lot with respect to our PosScan model, there is still information that could be added to improve fixation duration prediction.

The present work implemented an efficient and precise neural network model that predicts subject scanpaths, by designing and creating a way to feed our model with the most relevant information embedded in features. We believe that the proposed models could be further improved by including more information in the input, such as EEG data. Studies demonstrate that during free-viewing exploration, the brain occipital responses are time-locked to the onsets of saccades during visual search [Dandekar et al., 2012] and the brain occipital areas are time-modulated to the saccade onset rather than to the fixation onset [Devia et al., 2017].

Regarding future work, there are many possible lines of research that can be pursued from this study:

- Add EEG data to the architecture since it contains a priori information about when a saccade will be performed.

- Add a positional embedding to the inputs to fix the sequential problem in Attention layers.
- Apply the Vision Transformer [Dosovitskiy et al., 2020] to create an end-to-end architecture that learns to generate the saliency maps that are going to be used as inputs to the model.
- Modify the loss function by adding a term which has the time shift between predictions and ground truth scanpaths.
- Train and test models using only saccades as input trying to find an optimum that allows predicting where the saccade will land with the least amount of saccade data.
- Train a single model on a large amount of data obtained from several different subjects, and then fit the model to a specific subject a.k.a fine-tune the model.
- Add as a new feature the probability of saccade occurrence conditioned to the time that the fixation has lasted (hazard function).

The implementation code of this work will be publicly available on URL: `https://github.com/cjotade/ScanPaths`

Bibliography

[Adeli et al., 2017] Adeli, H., Vitu, F., and Zelinsky, G. J. (2017). A model of the superior colliculus predicts fixation locations during scene viewing and visual search. *Journal of Neuroscience*, 37(6):1453–1467.

[Amano and Foster, 2014] Amano, K. and Foster, D. H. (2014). Influence of local scene color on fixation position in visual search. *JOSA A*, 31(4):A254–A262.

[Anderson et al., 2015] Anderson, N. C., Anderson, F., Kingstone, A., and Bischof, W. F. (2015). A comparison of scanpath comparison methods. *Behavior research methods*, 47(4):1377–1392.

[Araujo et al., 2001] Araujo, C., Kowler, E., and Pavel, M. (2001). Eye movements during visual search: The costs of choosing the optimal path. *Vision Research*, 41(25-26):3613–3625.

[Assens et al., 2018] Assens, M., Giro-i Nieto, X., McGuinness, K., and O'Connor, N. E. (2018). PathGAN: Visual scanpath prediction with generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, page 0.

[Assens Reina et al., 2017] Assens Reina, M., Giro-i Nieto, X., McGuinness, K., and O'Connor, N. E. (2017). Saltinet: Scan-path prediction on 360 degree images using saliency volumes. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2331–2338.

[Atiya et al., 1999] Atiya, A. F., El-Shoura, S. M., Shaheen, S. I., and El-Sherif, M. S. (1999). A comparison between neural-network forecasting techniques-case study: river flow forecasting. *IEEE Transactions on neural networks*, 10(2):402–409.

[Boccignone and Ferraro, 2004] Boccignone, G. and Ferraro, M. (2004). Modelling gaze shift as a constrained random walk. *Physica A: Statistical Mechanics and its Applications*, 331(1-2):207–218.

[Box et al., 2015] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley Sons.

[Brockmann and Geisel, 2000] Brockmann, D. and Geisel, T. (2000). The ecology of gaze shifts. *Neurocomputing*, 32:643–650.

[Brown and Mariano, 1984] Brown, B. W. and Mariano, R. S. (1984). Residual-based procedures for prediction and estimation in a nonlinear simultaneous system. *Econometrica: Journal of the*

Econometric Society, pages 321–343.

[Bruce and Tsotsos, 2005] Bruce, N. and Tsotsos, J. (2005). Saliency based on information maximization. In *Advances in neural information processing systems*, pages 155–162.

[Bruce et al., 2016] Bruce, N. D. B., Catton, C., and Janjic, S. (2016). A deeper look at saliency: Feature contrast, semantics, and beyond. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 516–524.

[Carhart-Harris et al., 2014] Carhart-Harris, R. L., Leech, R., Hellyer, P. J., Shanahan, M., Feilding, A., Tagliazucchi, E., Chialvo, D. R., and Nutt, D. (2014). The entropic brain: a theory of conscious states informed by neuroimaging research with psychedelic drugs. *Frontiers in human neuroscience*, page 20.

[Castelhano and Henderson, 2007] Castelhano, M. S. and Henderson, J. M. (2007). Initial scene representations facilitate eye movement guidance in visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 33(4):753.

[Cerf et al., 2008] Cerf, M., Harel, J., Einhäuser, W., and Koch, C. (2008). Predicting human gaze using low-level saliency combined with face detection. *Advances in neural information processing systems*, 20:1–7.

[Chua et al., 2005] Chua, H. F., Boland, J. E., and Nisbett, R. E. (2005). Cultural variation in eye movements during scene perception. *Proceedings of the National Academy of Sciences*, 102(35):12629–12633.

[Clark, 2013] Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204.

[Clarke et al., 2017] Clarke, A. D. F., Stainer, M. J., Tatler, B. W., and Hunt, A. R. (2017). The saccadic flow baseline: Accounting for image-independent biases in fixation behavior. *Journal of vision*, 17(11):12.

[Connor et al., 2004] Connor, C. E., Egeth, H. E., and Yantis, S. (2004). Visual attention: bottom-up versus top-down. *Current biology*, 14(19):R850–R852.

[Coutrot et al., 2018] Coutrot, A., Hsiao, J. H., and Chan, A. B. (2018). Scanpath modeling and classification with hidden Markov models. *Behavior research methods*, 50(1):362–379.

[Cristino et al., 2010] Cristino, F., Mathôt, S., Theeuwes, J., and Gilchrist, I. D. (2010). ScanMatch: A novel method for comparing fixation sequences. *Behavior research methods*, 42(3):692–700.

[Dandekar et al., 2012] Dandekar, S., Privitera, C., Carney, T., and Klein, S. A. (2012). Neural saccadic response estimation during natural viewing. *Journal of neurophysiology*, 107(6):1776–1790.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision*

and pattern recognition, pages 248–255. Ieee.

[Devia et al., 2017] Devia, C., Montefusco-Siegmund, R., Egaña, J. I., and Maldonado, P. E. (2017). Precise timing of sensory modulations coupled to eye movements during active vision. *bioRxiv*, page 144477.

[Dewhurst et al., 2012] Dewhurst, R., Nyström, M., Jarodzka, H., Foulsham, T., Johansson, R., and Holmqvist, K. (2012). It depends on how you look at it: Scanpath comparison in multiple dimensions with MultiMatch, a vector-based approach. *Behavior research methods*, 44(4):1079–1100.

[Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., and Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[Engbert et al., 2015] Engbert, R., Trukenbrod, H. A., Barthelmé, S., and Wichmann, F. A. (2015). Spatial statistics and attentional dynamics in scene viewing. *Journal of vision*, 15(1):14.

[Fahimi and Bruce, 2021] Fahimi, R. and Bruce, N. D. B. (2021). On metrics for measuring scanpath similarity. *Behavior Research Methods*, 53(2):609–628.

[Friston, 2003] Friston, K. (2003). Learning and inference in the brain. *Neural Networks*, 16(9):1325–1352.

[Friston, 2005] Friston, K. (2005). A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836.

[Friston, 2009] Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13(7):293–301.

[Friston, 2013] Friston, K. (2013). Life as we know it. *Journal of the Royal Society Interface*, 10(86):20130475.

[Friston et al., 2006] Friston, K., Kilner, J., and Harrison, L. (2006). A free energy principle for the brain. *Journal of physiology-Paris*, 100(1-3):70–87.

[Gal and Ghahramani, 2016a] Gal, Y. and Ghahramani, Z. (2016a). A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems*, 29:1019–1027.

[Gal and Ghahramani, 2016b] Gal, Y. and Ghahramani, Z. (2016b). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

[Geurts et al., 2006] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

[Girard and Berthoz, 2005] Girard, B. and Berthoz, A. (2005). From brainstem to cortex: com-

putational models of saccade generation circuitry. *Progress in neurobiology*, 77(4):215–251.

[Girshick, 2015] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press. `http://www.deeplearningbook.org`.

[Granger and Newbold, 1976] Granger, C. W. J. and Newbold, P. (1976). Forecasting transformed series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 38(2):189–203.

[Hamzaçebi et al., 2009] Hamzaçebi, C., Akay, D., and Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert systems with applications*, 36(2):3839–3844.

[Harel et al., 2006] Harel, J., Koch, C., and Perona, P. (2006). Graph-based visual saliency. *Advances in neural information processing systems*, 19.

[Helo et al., 2014] Helo, A., Pannasch, S., Sirri, L., and Rämä, P. (2014). The maturation of eye movement behavior: Scene viewing characteristics in children and adults. *Vision research*, 103:83–91.

[Henderson, 2003] Henderson, J. M. (2003). Human gaze control during real-world scene perception. *Trends in cognitive sciences*, 7(11):498–504.

[Henderson and Hollingworth, 1998] Henderson, J. M. and Hollingworth, A. (1998). Eye movements during scene viewing: An overview. *Eye guidance in reading and scene perception*, pages 269–293.

[Henderson et al., 1999] Henderson, J. M., Weeks Jr, P. A., and Hollingworth, A. (1999). The effects of semantic consistency on eye movements during complex scene viewing. *Journal of experimental psychology: Human perception and performance*, 25(1):210.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hosoya et al., 2005] Hosoya, T., Baccus, S. A., and Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, 436(7047):71–77.

[Hu et al., 2020] Hu, Z., Li, S., and Gai, M. (2020). Temporal continuity of visual attention for future gaze prediction in immersive virtual reality. *Virtual Reality and Intelligent Hardware*, 2(2):142–152.

[Huang et al., 2015] Huang, X., Shen, C., Boix, X., and Zhao, Q. (2015). Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 262–270.

[Islam et al., 2018] Islam, M. A., Kalash, M., and Bruce, N. D. B. (2018). Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7142–7150.

[Islam et al., 2017] Islam, M. A., Kalash, M., Rochan, M., Bruce, N. D. B., and Wang, Y. (2017). Salient Object Detection using a Context-Aware Refinement Network. In *BMVC*.

[Itti, 2000] Itti, L. (2000). *Models of bottom-up and top-down visual attention*. California Institute of Technology.

[Itti and Koch, 2000] Itti, L. and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10-12):1489–1506.

[Itti et al., 1998] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259.

[Jarodzka et al., 2010] Jarodzka, H., Holmqvist, K., and Nyström, M. (2010). A vector-based, multidimensional scanpath similarity measure. In *Proceedings of the 2010 symposium on eye-tracking research applications*, pages 211–218.

[Kalesnykas and Sparks, 1996] Kalesnykas, R. P. and Sparks, D. L. (1996). REVIEW: The Primate Superior Colliculus and the Control of Saccadic Eye Movements. *The Neuroscientist*, 2(5):284–292.

[Kienzle et al., 2007] Kienzle, W., Wichmann, F. A., Schölkopf, B., and Franz, M. O. (2007). A nonparametric approach to bottom-up visual saliency. *Advances in neural information processing systems*, 19:689.

[Kline, 2004] Kline, D. M. (2004). Methods for multi-step time series forecasting neural networks. In *Neural networks in business forecasting*, pages 226–250. IGI Global.

[Koch and Ullman, 1985] Koch, C. and Ullman, S. (1985). Shifts in selective visual attention: towards the. *Human neurobiology*, 4:219–227.

[Krauzlis et al., 2013] Krauzlis, R. J., Lovejoy, L. P., and Zénon, A. (2013). Superior colliculus and visual spatial attention. *Annual review of neuroscience*, 36:165–182.

[Kümmerer and Bethge, 2021] Kümmerer, M. and Bethge, M. (2021). State-of-the-Art in Human Scanpath Prediction. *arXiv preprint arXiv:2102.12239*.

[Kümmerer et al., 2016] Kümmerer, M., Wallis, T. S. A., and Bethge, M. (2016). DeepGaze II: Reading fixations from deep features trained on object recognition. *arXiv preprint arXiv:1610.01563*.

[Kummerer et al., 2017] Kummerer, M., Wallis, T. S. A., Gatys, L. A., and Bethge, M. (2017). Understanding low-and high-level contributions to fixation prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4789–4798.

[Lang, 2005] Lang, P. J. (2005). International affective picture system (IAPS): Affective ratings of pictures and instruction manual. *Technical report*.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

[Lin and Granger, 1994] Lin, J. and Granger, C. W. J. (1994). Forecasting from non-linear models in practice. *Journal of Forecasting*, 13(1):1–9.

[Linardos et al., 2021] Linardos, A., Kümmerer, M., Press, O., and Bethge, M. (2021). DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12919–12928.

[Liu et al., 2013] Liu, H., Xu, D., Huang, Q., Li, W., Xu, M., and Lin, S. (2013). Semantically-based human scanpath estimation with hmms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3232–3239.

[Loftus, 1985] Loftus, G. R. (1985). Picture perception: Effects of luminance on available information and information-extraction rate. *Journal of Experimental Psychology: General*, 114(3):342.

[Loftus et al., 1992] Loftus, G. R., Kaufman, L., Nishimoto, T., and Ruthruff, E. (1992). Effects of visual degradation on eye-fixation duration, perceptual processing, and long-term visual memory. In *Eye movements and visual cognition*, pages 203–226. Springer.

[Mannan et al., 1996] Mannan, S. K., Ruddock, K. H., and Wooding, D. S. (1996). The relationship between the locations of spatial features and those of fixations made during visual examination of briefly presented images. *Spatial vision*.

[Menze et al., 2009] Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., and Hamprecht, F. A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10(1):1–16.

[Najemnik and Geisler, 2008] Najemnik, J. and Geisler, W. S. (2008). Eye movement statistics in humans are consistent with an optimal search strategy. *Journal of Vision*, 8(3):4.

[Najemnik and Geisler, 2009] Najemnik, J. and Geisler, W. S. (2009). Simple summation rule for optimal fixation selection in visual search. *Vision research*, 49(10):1286–1294.

[Needleman and Wunsch, 1970] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

[Onat et al., 2014] Onat, S., Açık, A., Schumann, F., and König, P. (2014). The contributions of image content and behavioral relevancy to overt attention. *PLoS One*, 9(4):e93254.

[Pannasch et al., 2008] Pannasch, S., Helmert, J. R., Roth, K., Herbold, A.-K., and Walter, H. (2008). Visual fixation durations and saccade amplitudes: Shifting relationship in a variety of conditions. *Journal of Eye Movement Research*, 2(2).

[Parkhurst et al., 2002] Parkhurst, D., Law, K., and Niebur, E. (2002). Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1):107–123.

[Perry and Geisler, 2002] Perry, J. S. and Geisler, W. S. (2002). Gaze-contingent real-time simulation of arbitrary visual fields. In *Human vision and electronic imaging VII*, volume 4662, pages 57–69. International Society for Optics and Photonics.

[Pinto et al., 2013] Pinto, Y., van der Leij, A. R., Sligte, I. G., Lamme, V. A. F., and Scholte, H. S. (2013). Bottom-up and top-down attention are independent. *Journal of vision*, 13(3):16.

[Piotrowski and Campbell, 1982] Piotrowski, L. N. and Campbell, F. W. (1982). A demonstration of the visual importance and flexibility of spatial-frequency amplitude and phase. *Perception*, 11(3):337–346.

[Rao and Ballard, 1999] Rao, R. P. N. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87.

[Rayner, 1998] Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372.

[Redmon and Farhadi, 2017] Redmon, J. and Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Rottach et al., 1997] Rottach, K. G., Von Maydell, R. D., Das, V. E., Zivotofsky, A. Z., Discenna, A. O., Gordon, J. L., Landis, D. M. D., and Leigh, R. J. (1997). Evidence for independent feedback control of horizontal and vertical saccades from Niemann-Pick type C disease. *Vision research*, 37(24):3627–3638.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[Salthouse and Ellis, 1980] Salthouse, T. A. and Ellis, C. L. (1980). Determinants of eye-fixation duration. *The American journal of psychology*, pages 207–234.

[Sandler et al., 2018] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

[Schütt et al., 2017] Schütt, H. H., Rothkegel, L. O. M., Trukenbrod, H. A., Reich, S., Wichmann, F. A., and Engbert, R. (2017). Likelihood-based parameter estimation and comparison of dynamical cognitive models. *Psychological review*, 124(4):505.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Sorjamaa et al., 2007] Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., and Lendasse, A. (2007). Methodology for long-term prediction of time series. *Neurocomputing*, 70(16-18):2861–2869.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

[Staub et al., 2012] Staub, A., Abbott, M., and Bogartz, R. S. (2012). Linguistically guided anticipatory eye movements in scene viewing. *Visual Cognition*, 20(8):922–946.

[Sun et al., 2014] Sun, X., Yao, H., Ji, R., and Liu, X.-M. (2014). Toward statistical modeling of saccadic eye-movement and visual saliency. *IEEE Transactions on Image Processing*, 23(11):4649–4662.

[Taieb, 2014] Taieb, S. B. (2014). Machine learning strategies for multi-step-ahead time series forecasting. *Universit Libre de Bruxelles, Belgium*, pages 75–86.

[Treisman and Gelade, 1980] Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136.

[Van Diepen et al., 1995] Van Diepen, P. M. J., De Graef, P., and D'Ydewalle, G. (1995). Chronometry of foveal information extraction during scene perception. In *Studies in visual information processing*, volume 6, pages 349–362. Elsevier.

[Van Diepen et al., 1998] Van Diepen, P. M. J., Wampers, M., and D'Ydewalle, G. (1998). Functional division of the visual field: Moving masks and moving windows. In *Eye guidance in reading and scene perception*, pages 337–355. Elsevier.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Wang et al., 2018] Wang, C.-A., Huang, J., Yep, R., and Munoz, D. P. (2018). Comparing pupil light response modulation between saccade planning and working memory. *Journal of cognition*, 1(1).

[Wang et al., 2011] Wang, W., Chen, C., Wang, Y., Jiang, T., Fang, F., and Yao, Y. (2011). Simulating human saccadic scanpaths on natural images. In *CVPR 2011*, pages 441–448. IEEE.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[Xia et al., 2019] Xia, C., Han, J., Qi, F., and Shi, G. (2019). Predicting human saccadic scanpaths based on iterative representation learning. *IEEE Transactions on Image Processing*, 28(7):3502–3515.

[Xu et al., 2014] Xu, J., Jiang, M., Wang, S., Kankanhalli, M. S., and Zhao, Q. (2014). Predicting human gaze beyond pixels. *Journal of vision*, 14(1):28.

[Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

[Yang, 2017] Yang, X. (2017). Understanding the variational lower bound. *In: variational lower bound, ELBO, hard attention*, pages 1–4.

[Zanca et al., 2019] Zanca, D., Melacci, S., and Gori, M. (2019). Gravitational laws of focus of attention. *IEEE transactions on pattern analysis and machine intelligence*, 42(12):2983–2995.

[Zhang et al., 2008] Zhang, L., Tong, M. H., Marks, T. K., Shan, H., and Cottrell, G. W. (2008). SUN: A Bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32.

ANNEXES

Annexed A

PosScan architecture

A.1 Architecture parameters

LSTM Model from Figure 4.3:

- LSTM layer units: 30
- 1st Feed Forward layer units: 20
- 2nd Feed Forward layer units: 100
- Dropout: 0.2 for every layer
- Recurrent Dropout on LSTM layer: 0.2
- Learning rate: 0.0001
- Epochs: 500
- Patience (early stop): 15
- Optimizer: Adam
- Loss: MSE
- Input sequence length: 10

A.2 Metric results grouped by train image type

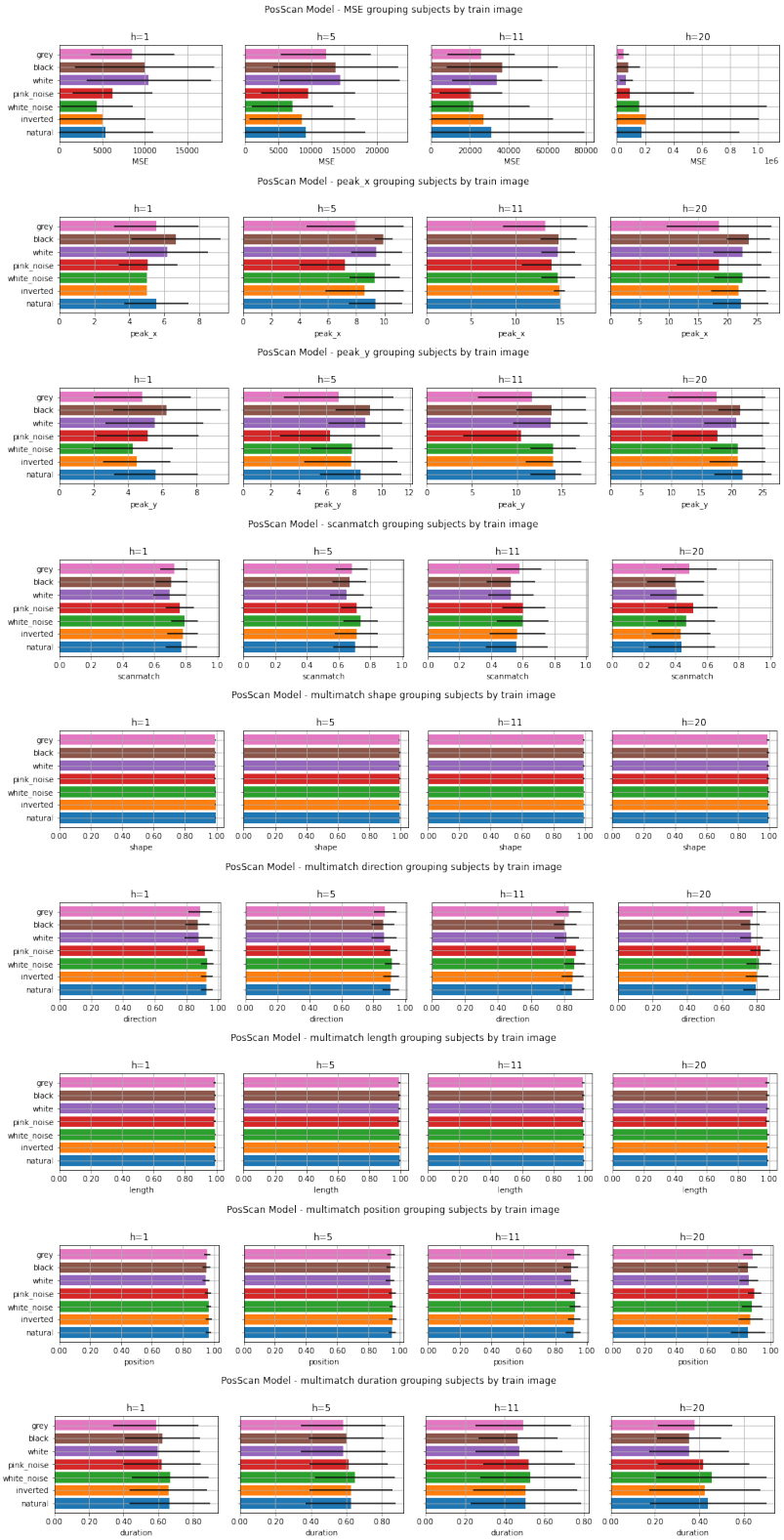

Figure A.1: General results for some metrics grouped by train image.

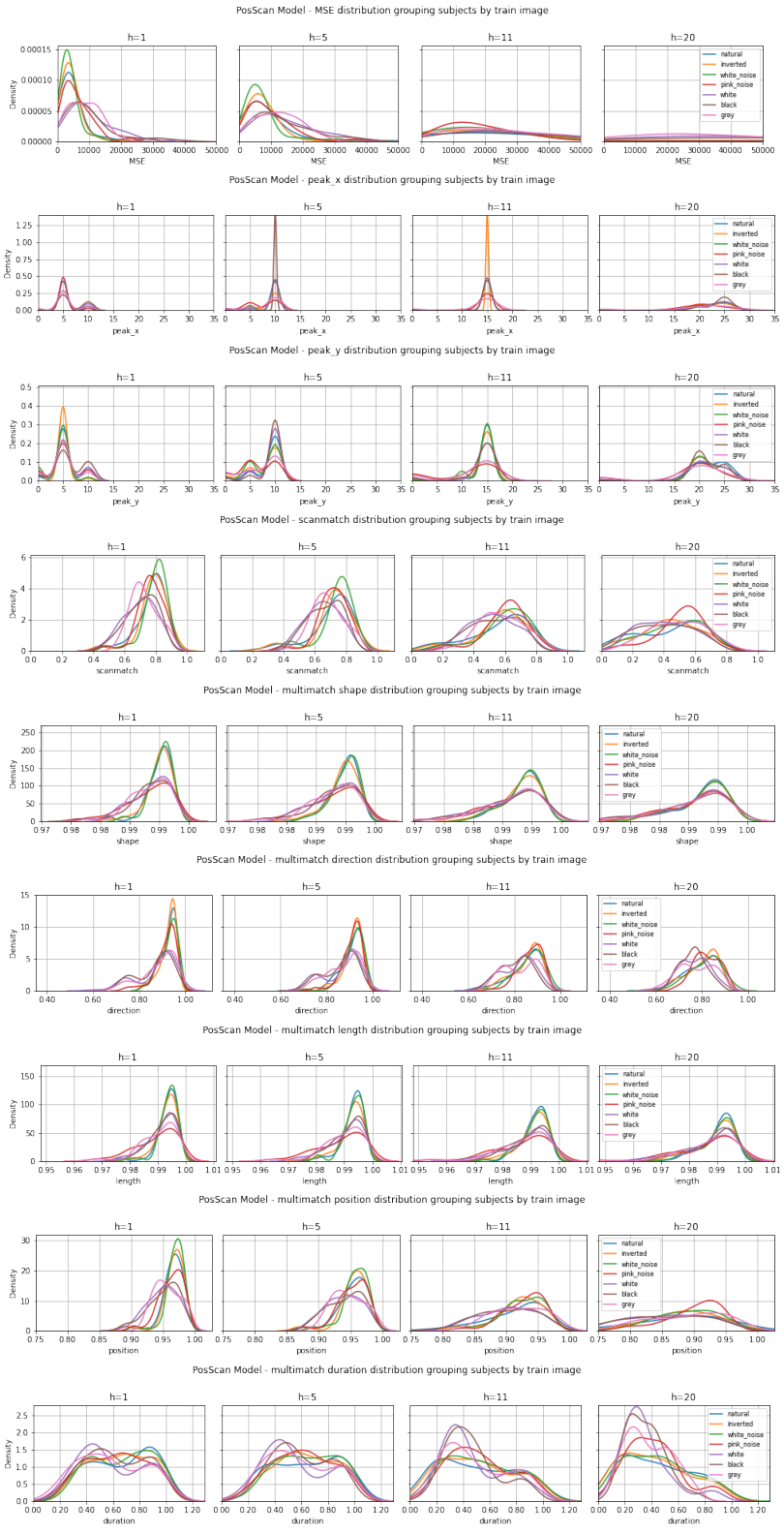

Figure A.2: General results for some metrics grouped by train image.

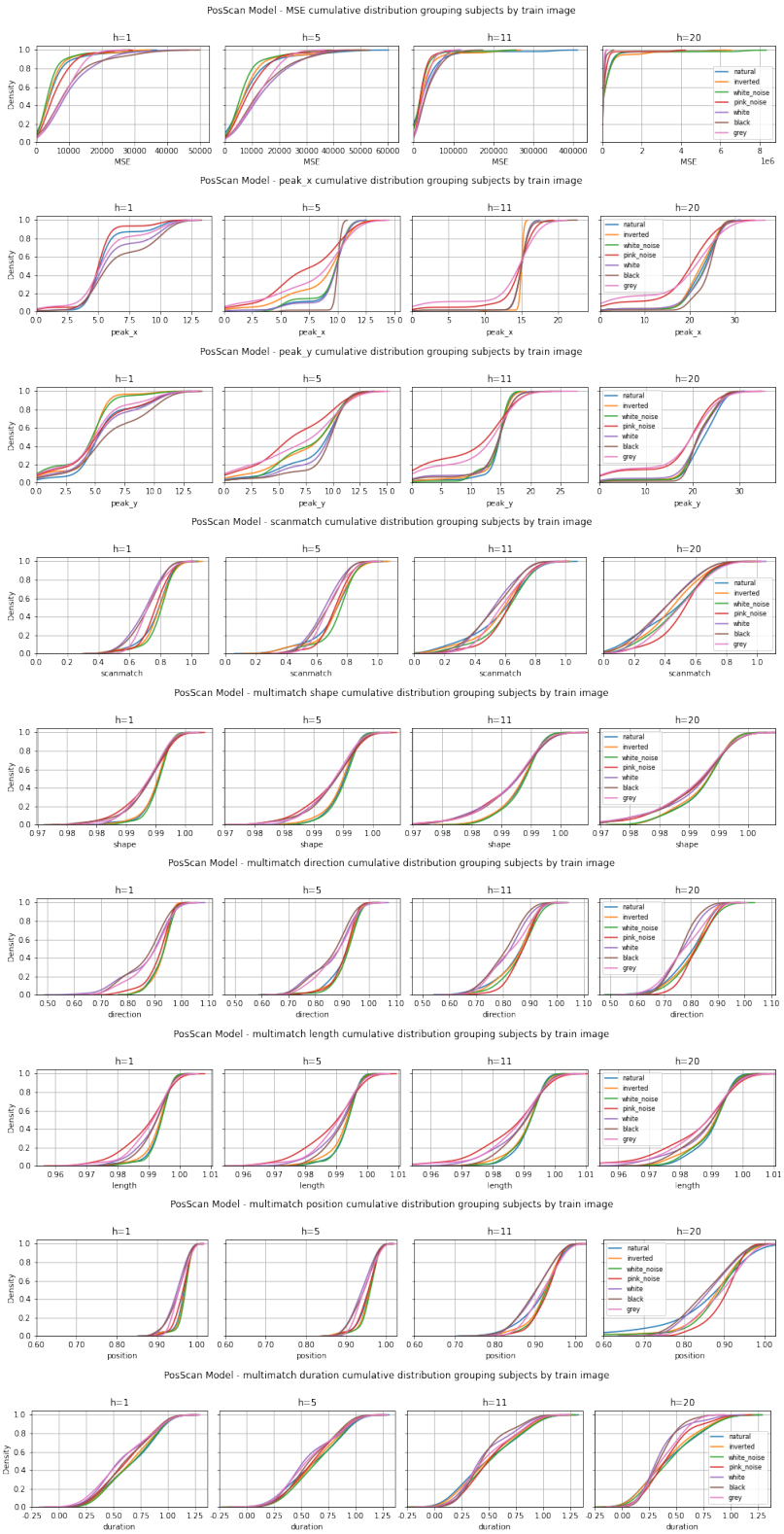

Figure A.3: General results for some metrics grouped by train image.

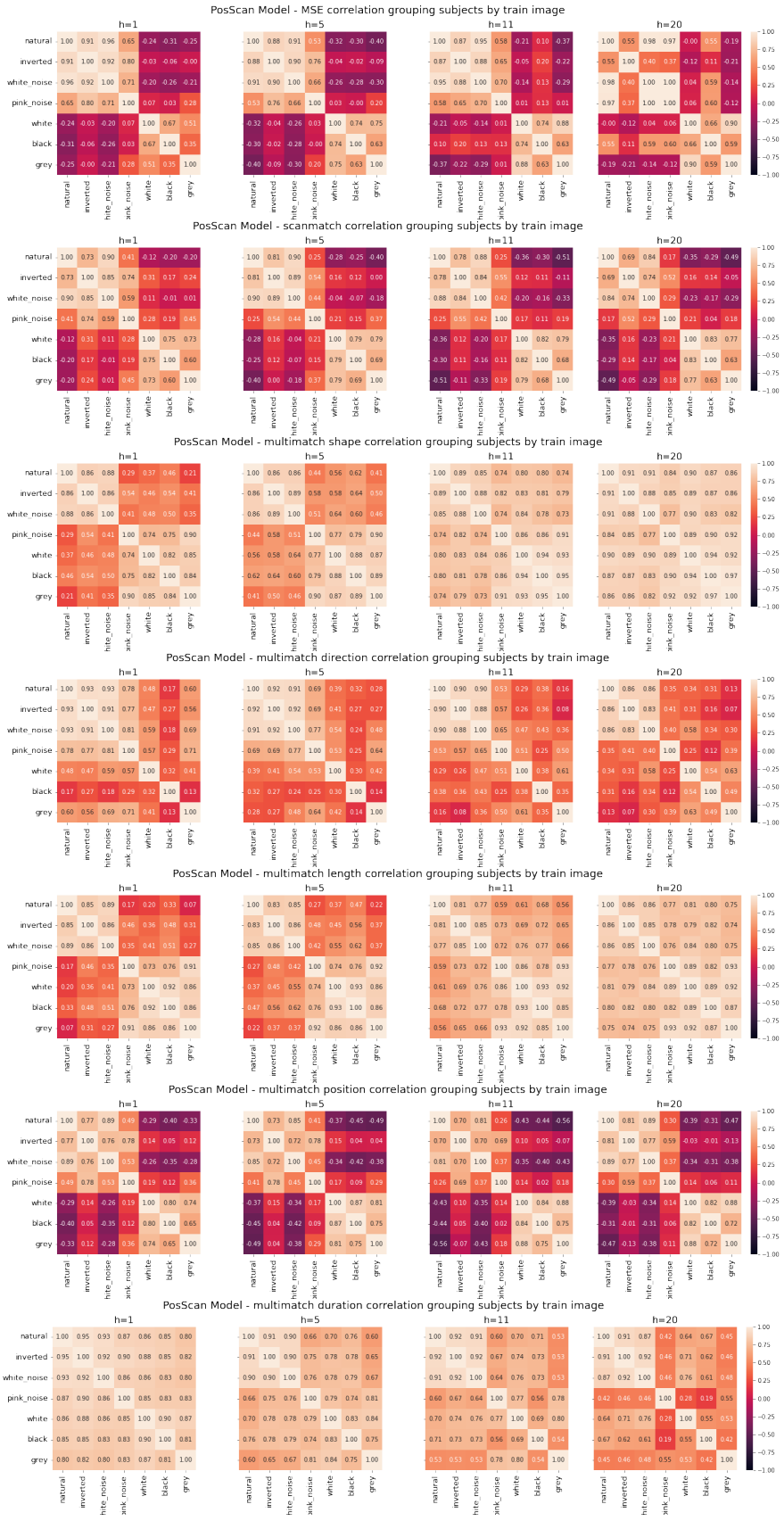

Figure A.4: Correlation results for some metrics grouped by train image.

A.3 Metric results grouped by predicted image type

Figure A.5: General results for some metrics grouped by predicted image.

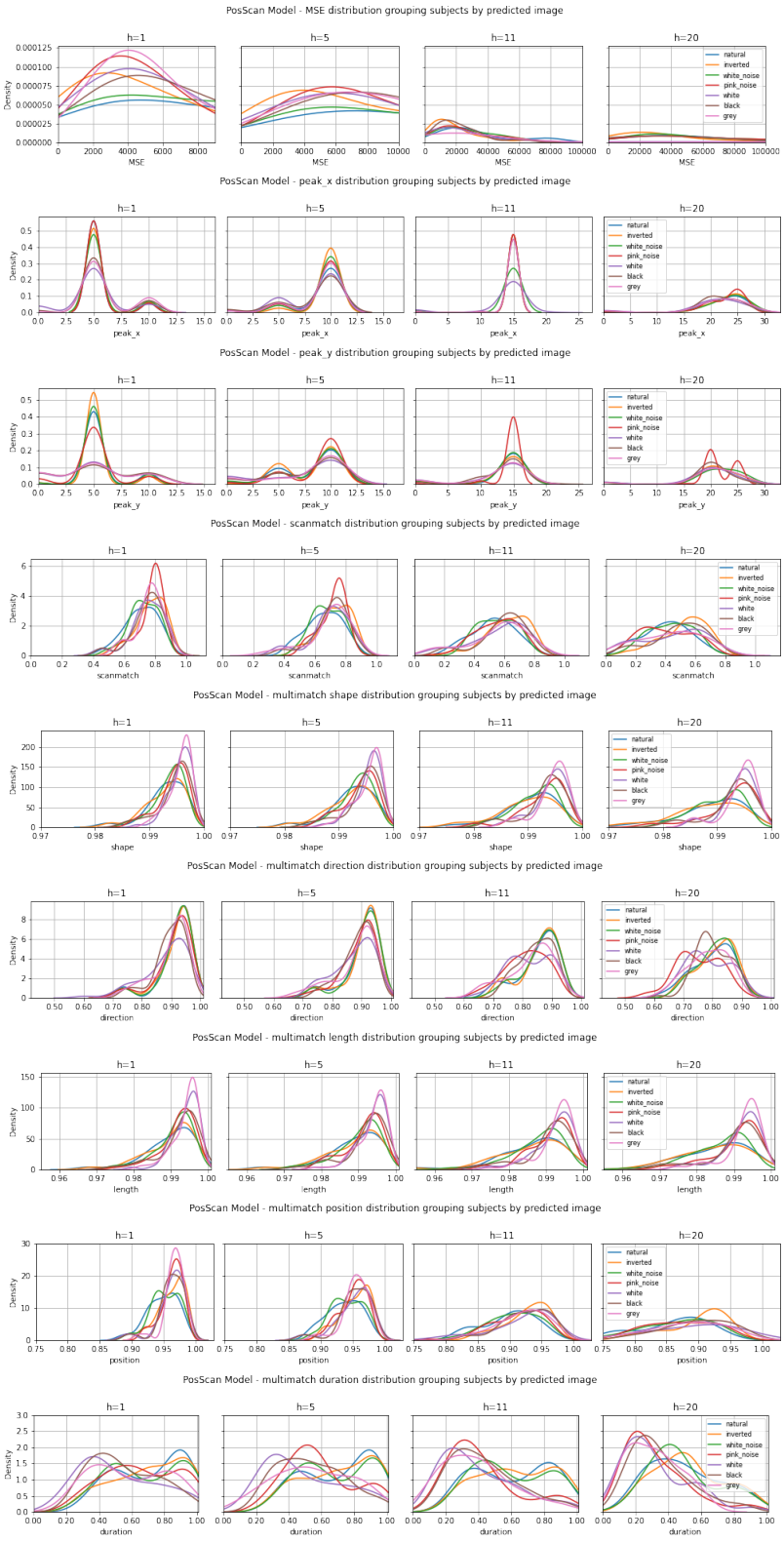

Figure A.6: General results for some metrics grouped by predicted image.

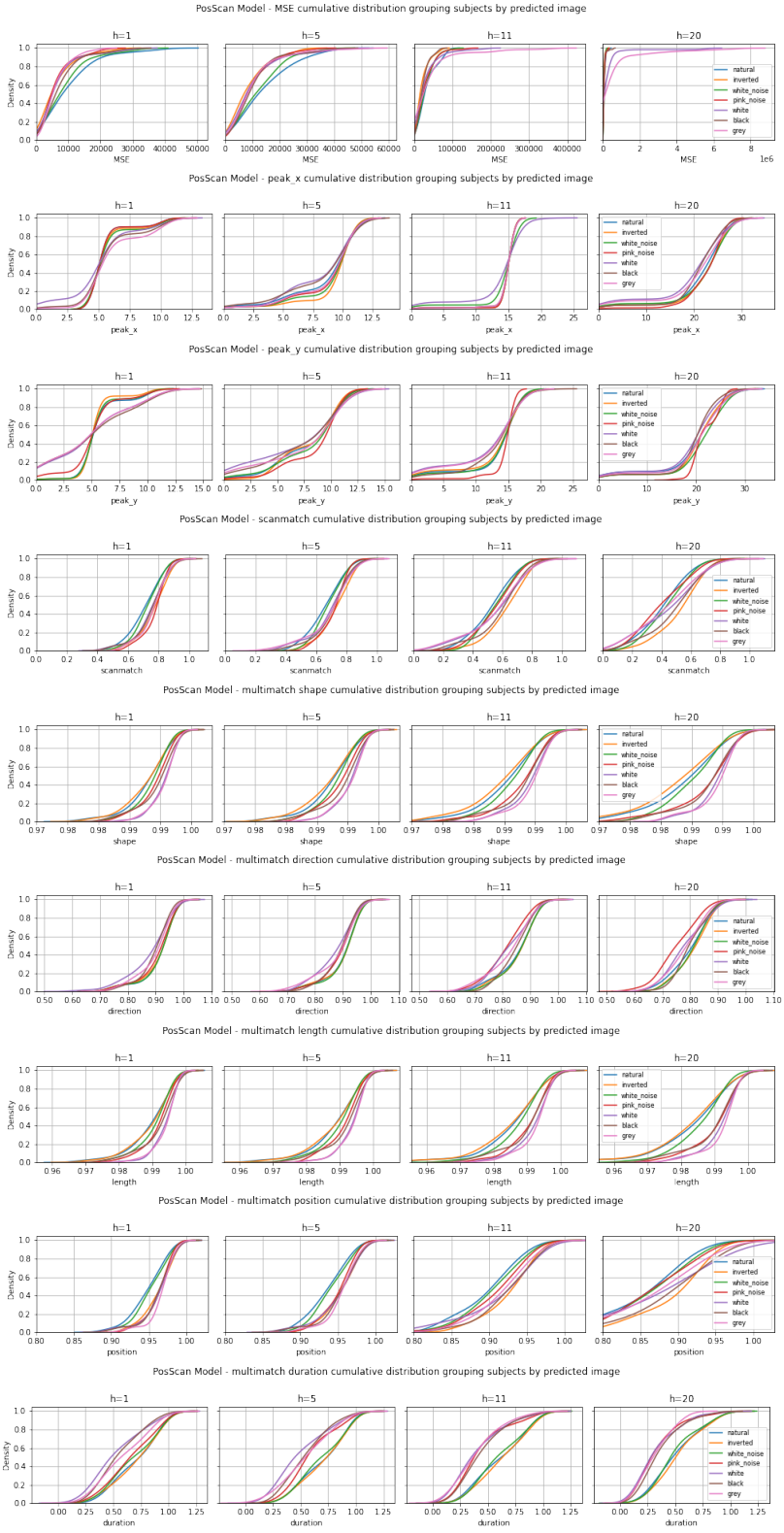

Figure A.7: General results for some metrics grouped by predicted image.

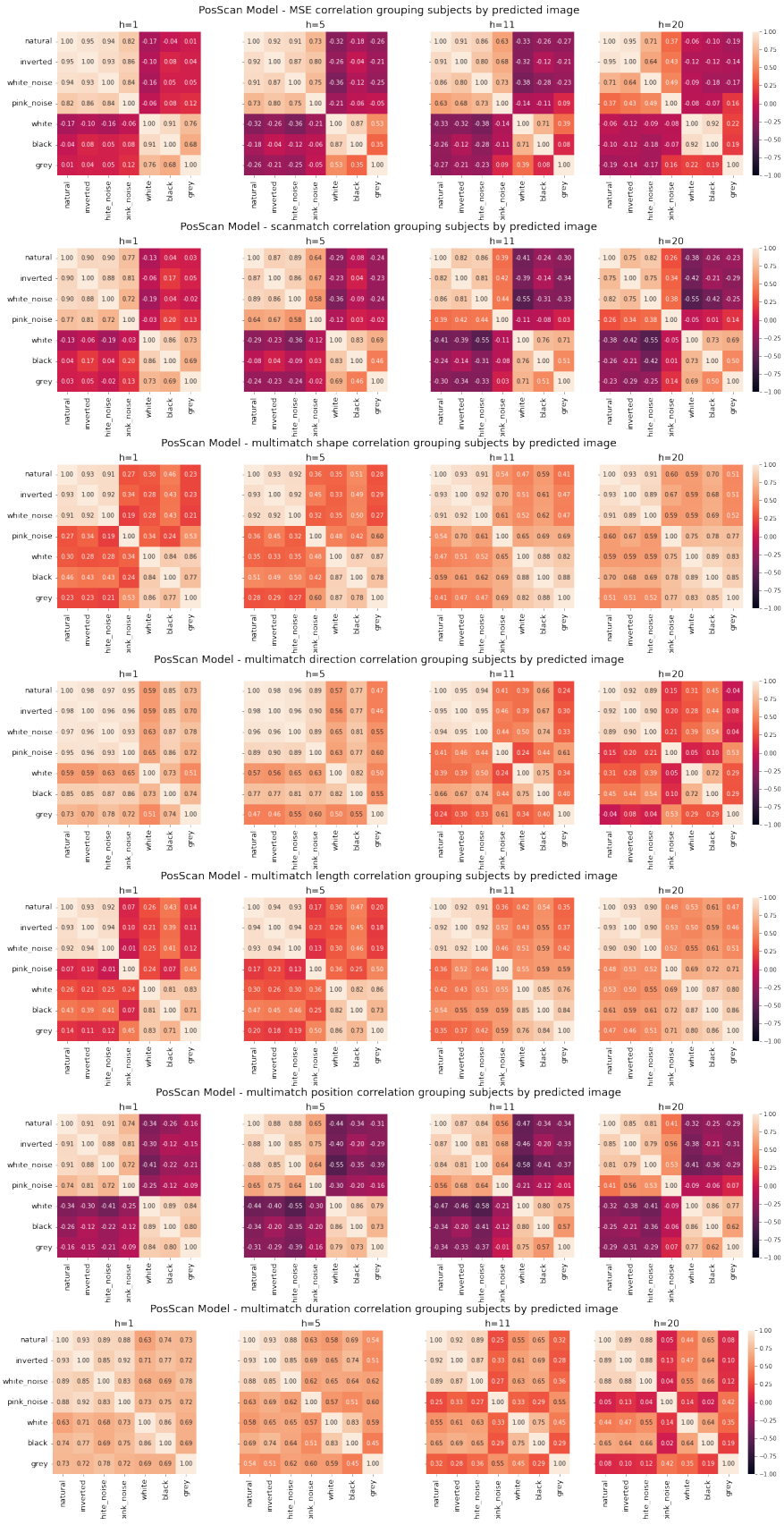

Figure A.8: Correlation results for some metrics grouped by predicted image.

Annexed B

FovSOS-FS architecture

B.1 Architecture parameters

Attention Model from Figure 4.25:

- Multi-Head Attention number of heads: 5
- Multi-Head Attention size of each attention head for query and key: 5
- 1st Feed Forward layer units: 80
- 2nd Feed Forward layer units: 100
- Dropout: 0.3 for every layer
- Dropout on Multi-Head Attention layer: 0.4
- Learning rate: 0.0001
- Epochs: 5000
- Patience (early stop): 150
- Optimizer: Adam
- Loss: MSE
- Input sequence length: 10
- PCA components selection percent: 95%

B.2 Metric results grouped by predicted image type

Figure B.1: General results for some metrics grouped by predicted image.

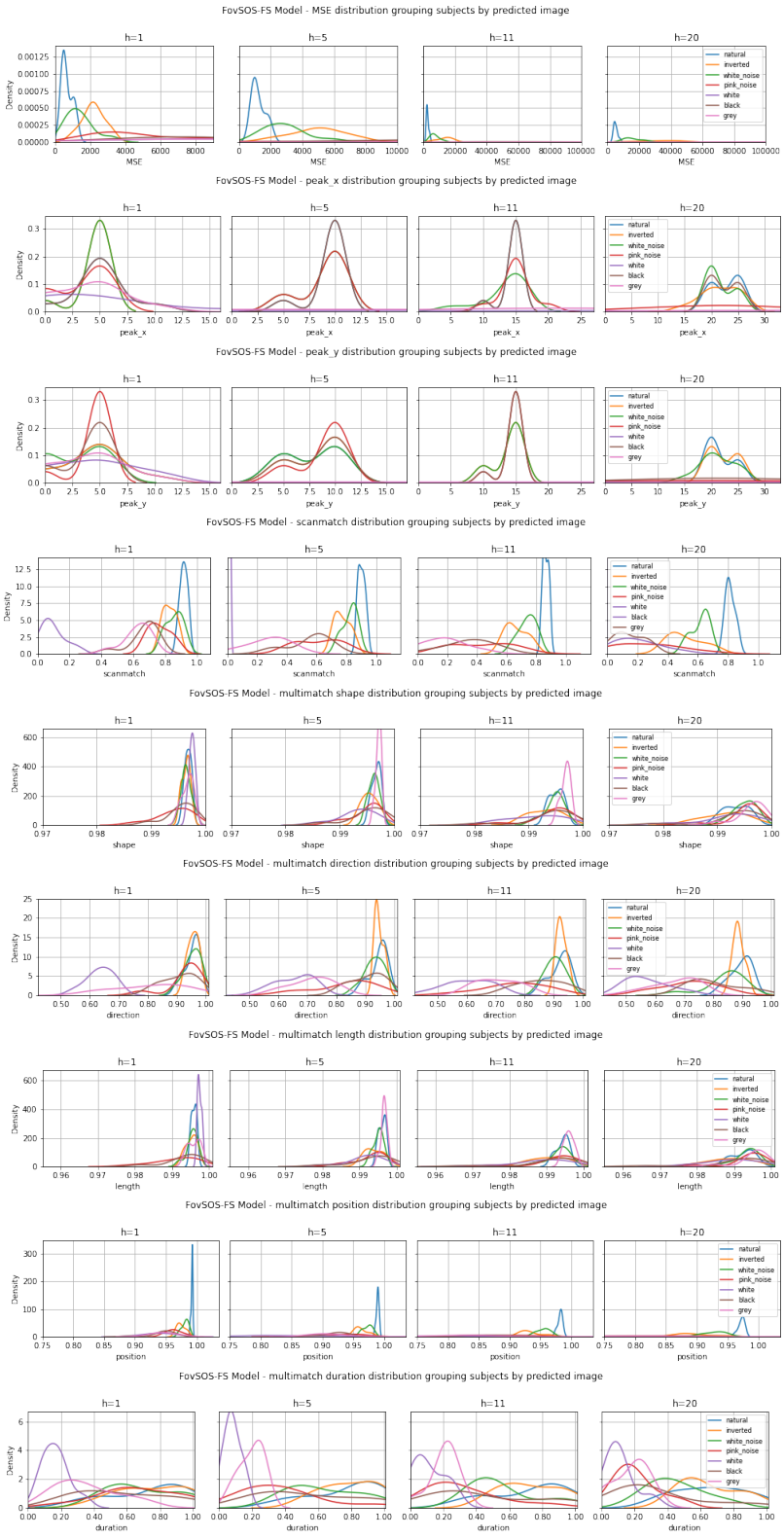

Figure B.2: General results for some metrics grouped by predicted image.

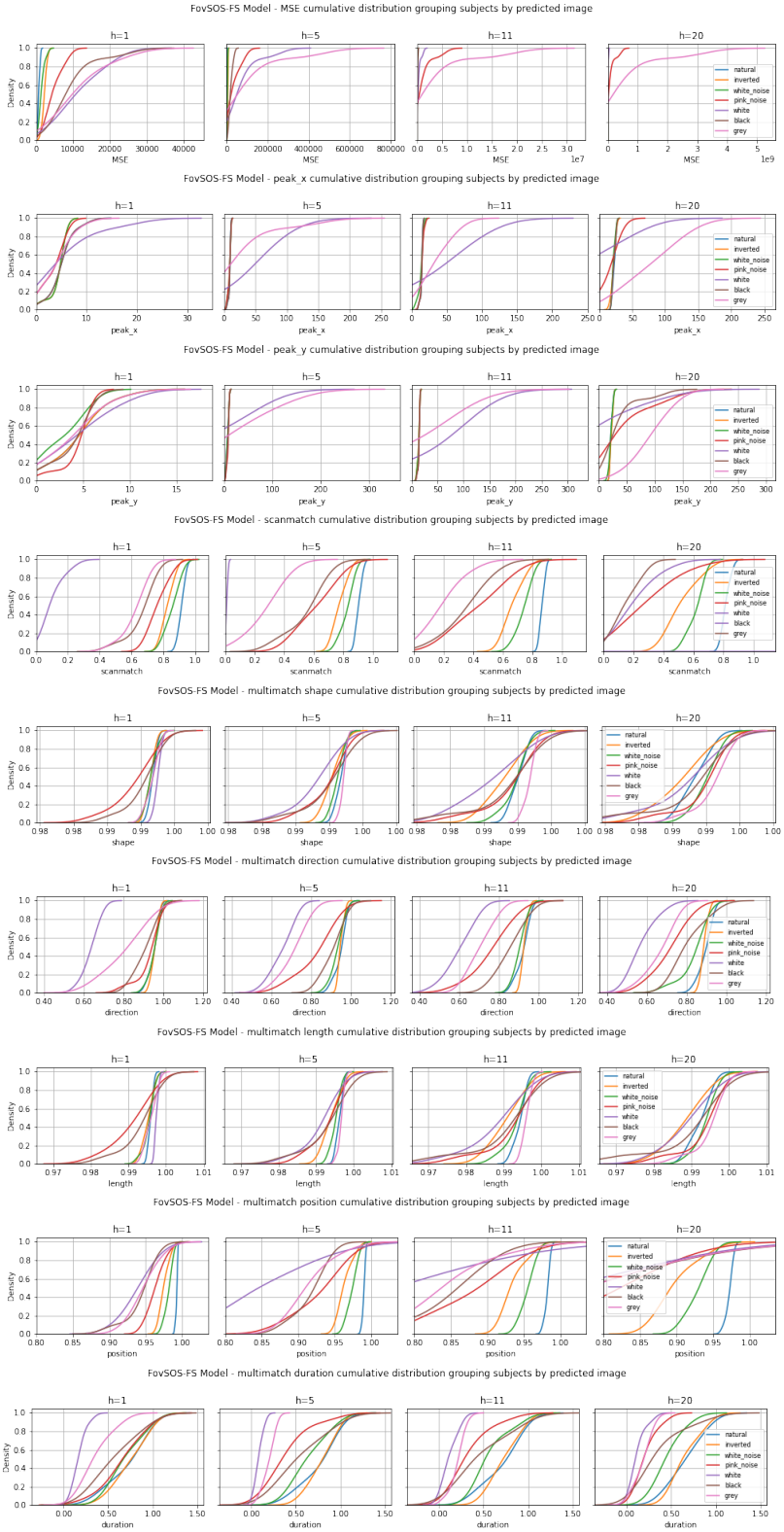

Figure B.3: General results for some metrics grouped by predicted image.

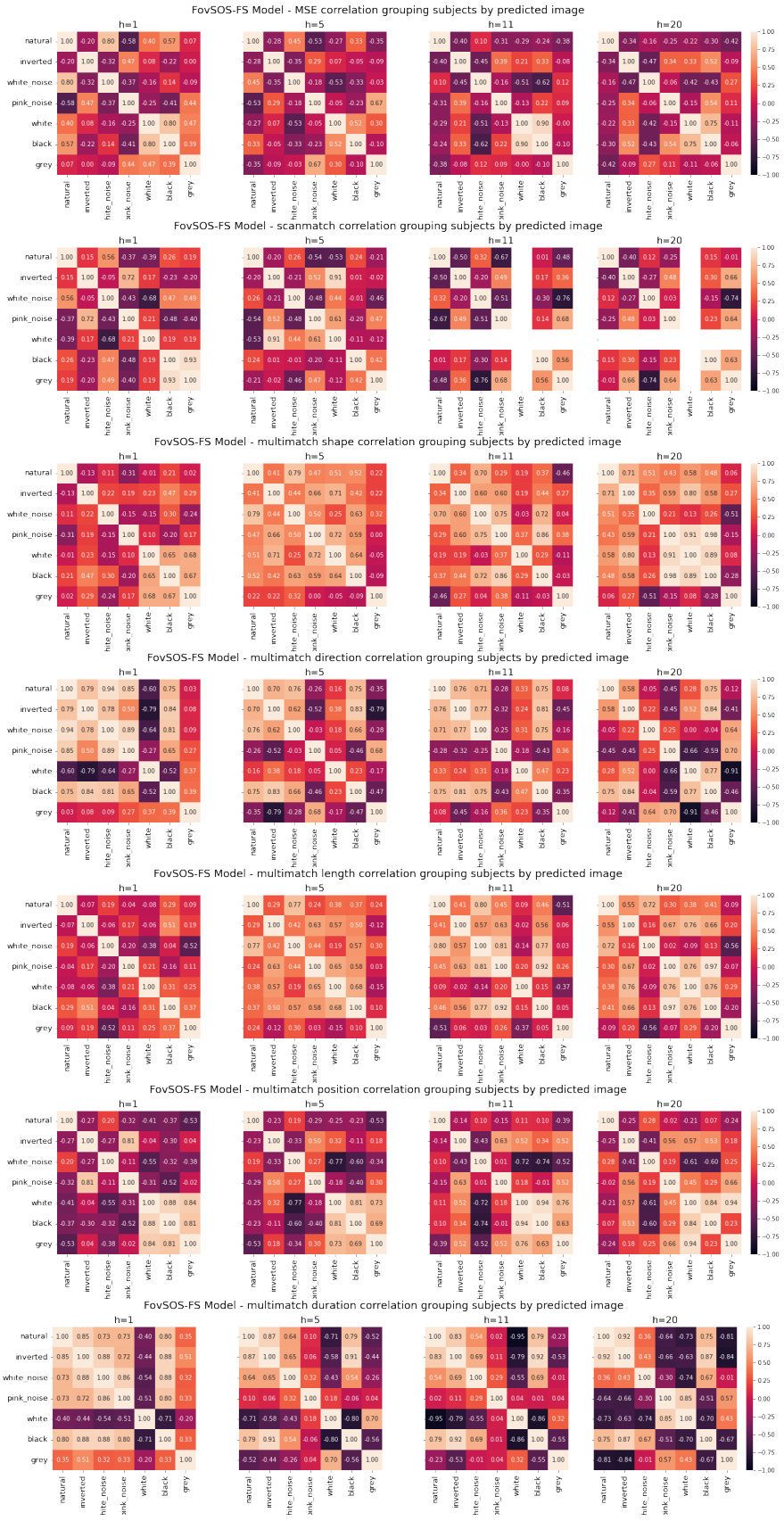

Figure B.4: Correlation results for some metrics grouped by predicted image.

Annexed C

FovSOS-FSD architecture

C.1 Architecture parameters

Attention Model from Figure 4.25:

- Multi-Head Attention number of heads: 5
- Multi-Head Attention size of each attention head for query and key: 5
- 1st Feed Forward layer units: 80
- 2nd Feed Forward layer units: 100
- Dropout: 0.3 for every layer
- Dropout on Multi-Head Attention layer: 0.4
- Learning rate: 0.0001
- Epochs: 5000
- Patience (early stop): 150
- Optimizer: Adam
- Loss: MSE
- Input sequence length: 10
- PCA components selection percent: 95%

C.2 Metric results grouped by train image type

Figure C.1: General results for some metrics grouped by train image.

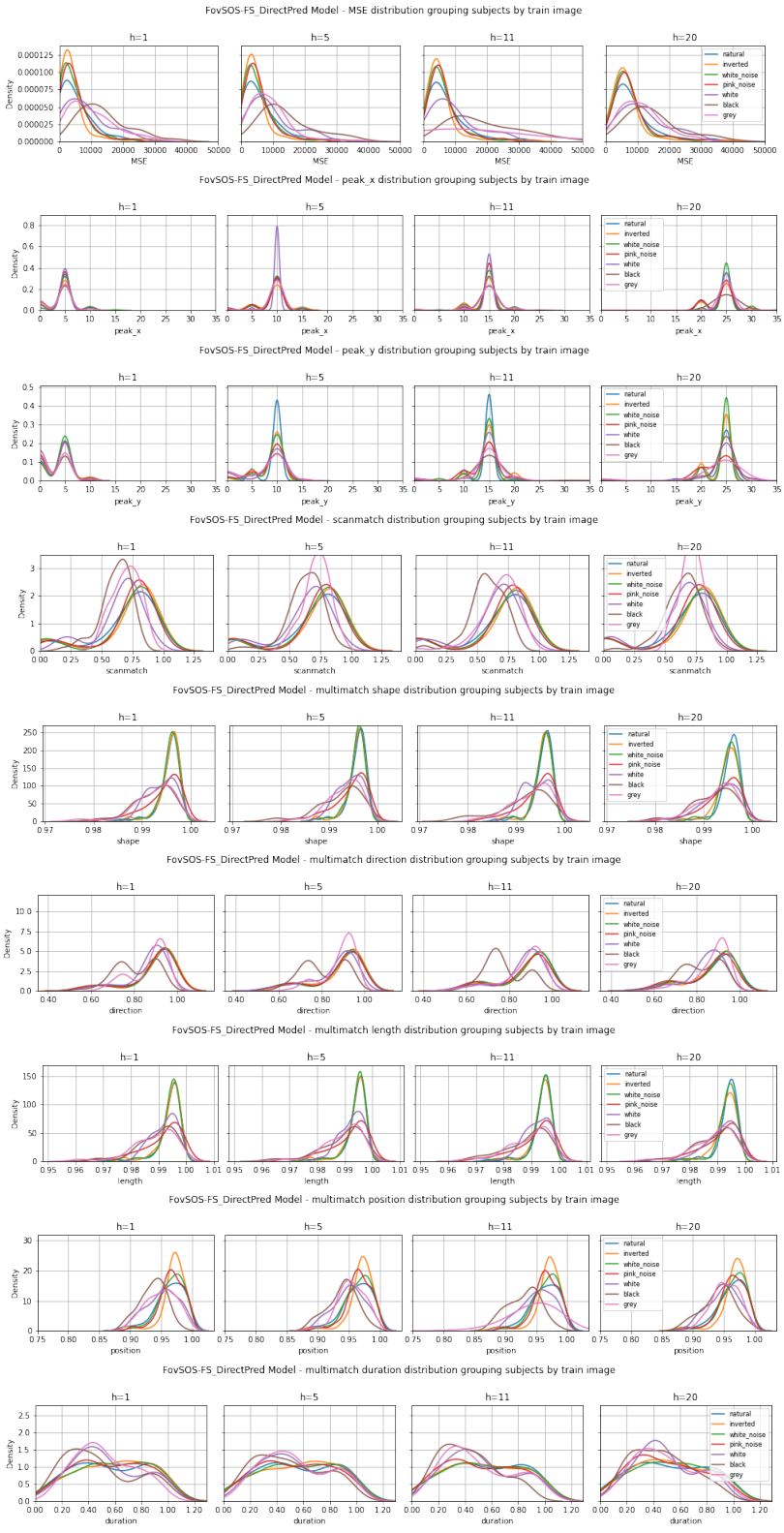

Figure C.2: General results for some metrics grouped by train image.

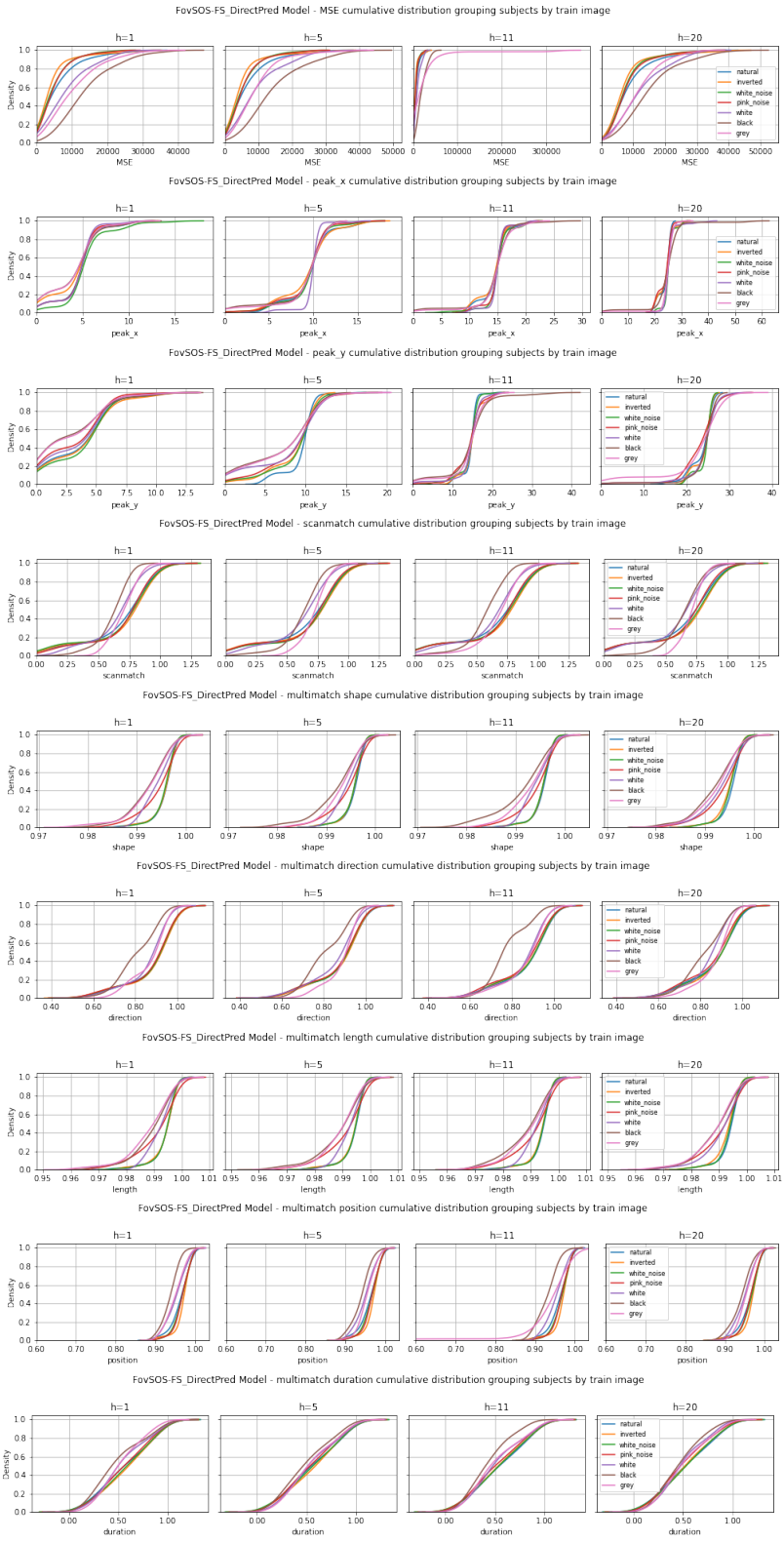

Figure C.3: General results for some metrics grouped by train image.

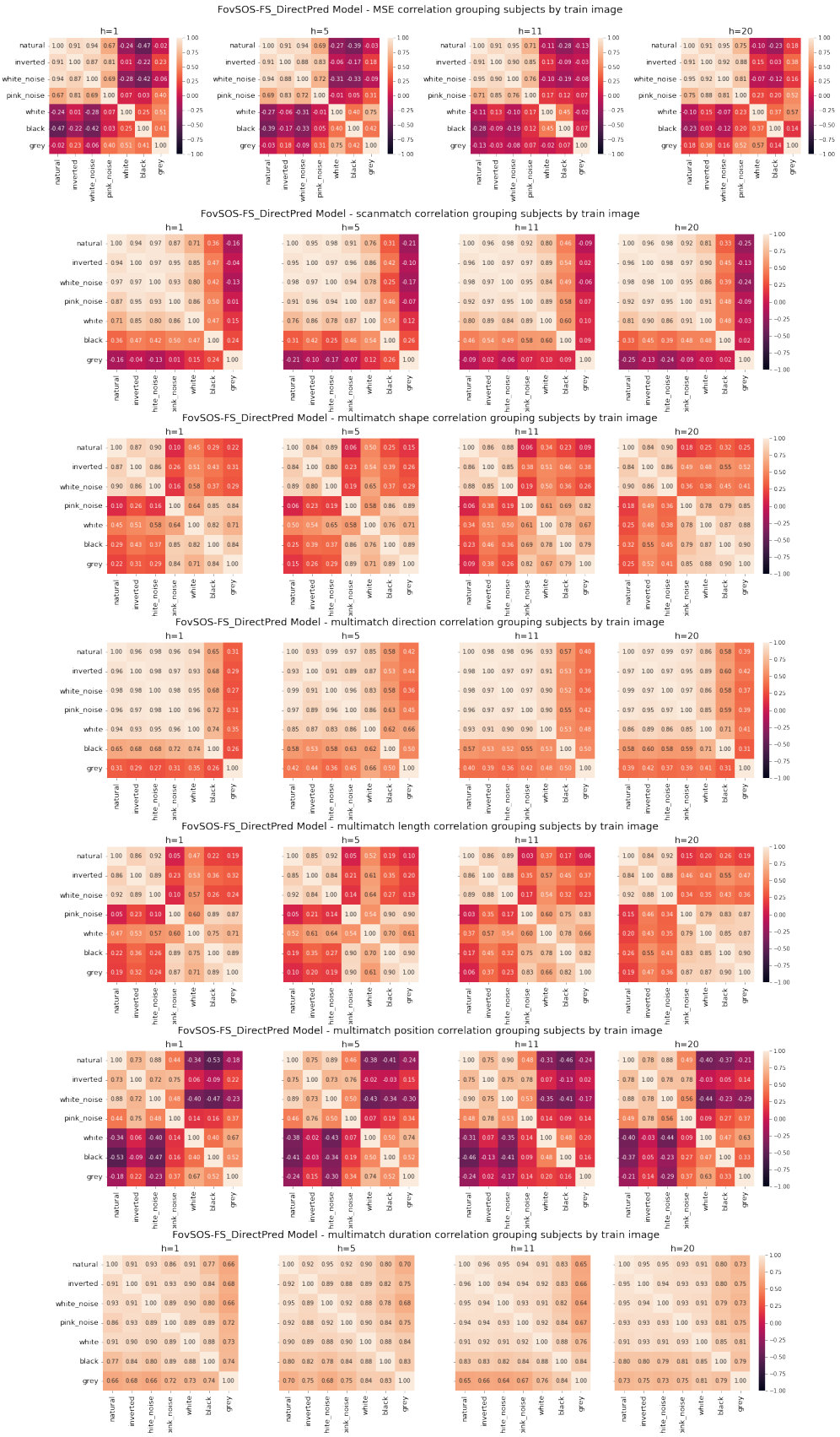

Figure C.4: Correlation results for some metrics grouped by train image.

C.3 Metric results grouped by predicted image type

Figure C.5: General results for some metrics grouped by predicted image.

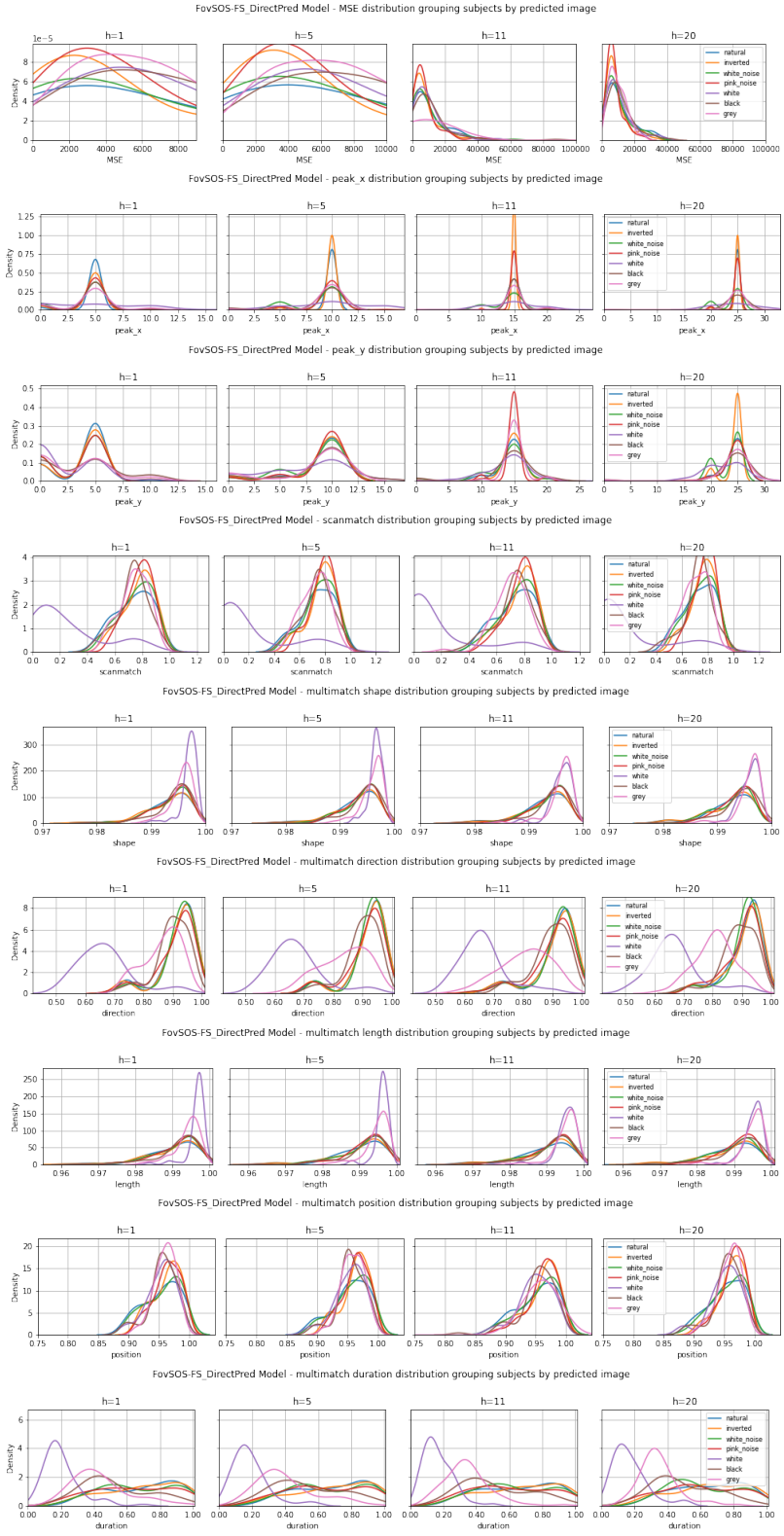

Figure C.6: General results for some metrics grouped by predicted image.

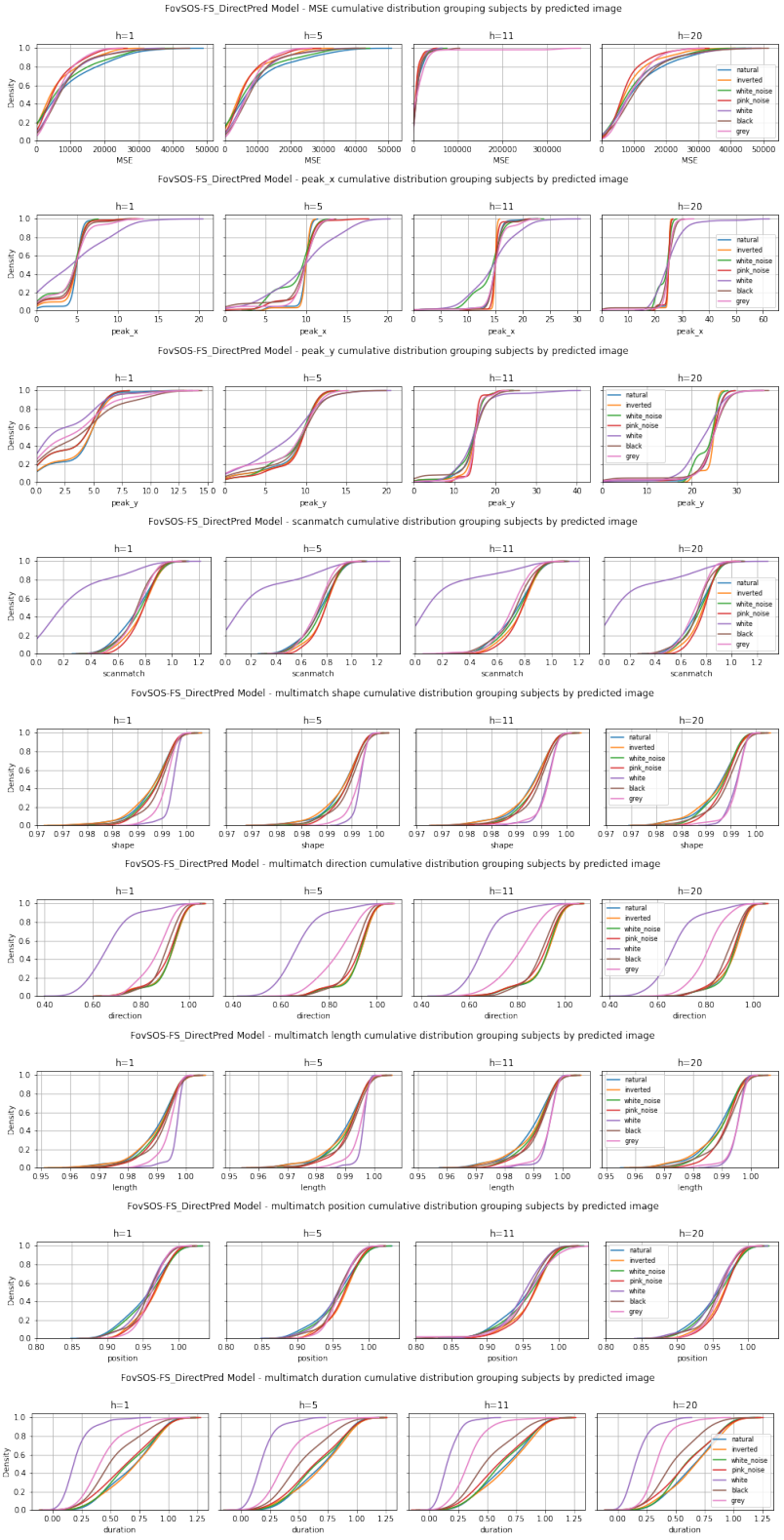

Figure C.7: General results for some metrics grouped by predicted image.

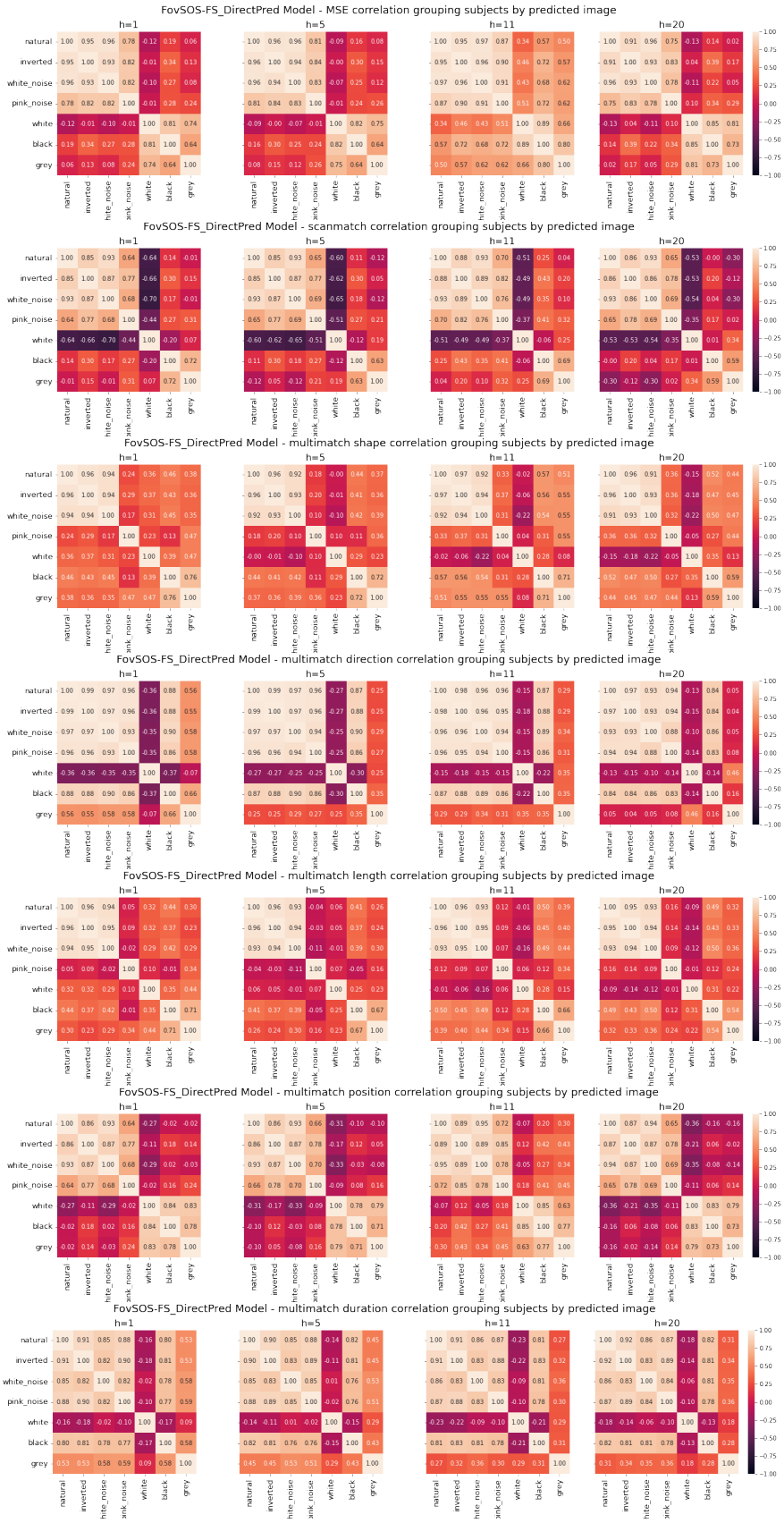

Figure C.8: Correlation results for some metrics grouped by predicted image.