Supervisor: Dr. Hantao Liu

CM3203: One Semester Individual Project

Total Credits: 40

# Analysis Of Machine Learning Models For Visual Saliency

Authored by

Nashalung Rai

# Content

# 1. Abstract

Detecting visually salient objects within images has been a challenge with machine learning models, often requiring feature extraction phases where humans have to intervene and manually extract the features within images.The advancements in deep learning have now made is possible to train neural networks with supervised or unsupervised learning, reducing the amount of time and energy required to label and extract features from the data. This project focuses on comparing different deep learning models for detecting visually salient objects within images. In order to highlight, the limitations of these models (VGG, ResNet and ML-Net) when it come to detecting salient objects when the input images are not of the same caliber as the training images.

# 2. Acknowledgements

I would like to give a big thank you to my supervisor, Dr. Hantao Liu for giving me the amazing opportunity to develop my skills within this particular field. His passion, knowledge, guidance and advice helped me hone my research and spur my interest to further pursue this area of research. Secondly, I would like to thank Miss Iris Zhao for being a very helpful research partner. Her inquisitive questions allowed me to think in a different perspective about various scientific questions. Last but not least, I would like to thank my family and friends for their encouragement and support throughout my whole degree and research project, as well as Miss Amanda Bali for believing in me throughout the whole project.

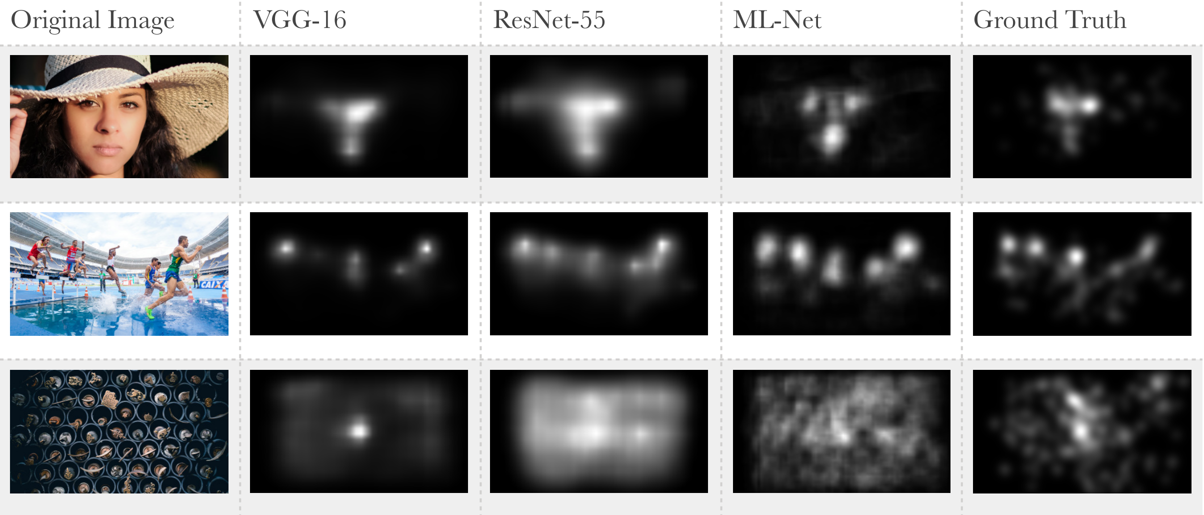
# 3. Introduction

Machine learning and neural networks have played a key role in allowing machines to analyse and understand salient objects within images and videos. Where a visually salient object refers to "distinct subjective qualities of items that makes them stand out" in images and videos which makes them attract human attention (Itti, 2007). The applications of which are made use in the fields of computer vision, robotics and also utilised by companies to aid advertisement of products. Breakthroughs in convolutional neural networks (CNN) which are used to process images have allowed researchers to analyse and predict salient objects within images by training these neural networks with input images along with ground truth saliency and fixation maps gathered through labour intensive data collection (Yang, Z. *et al.* 2014). Generally, these neural networks are trained using clean, high resolution images for training and testing, although this does yield successful results. It is still very unclear on their ability to predict salient objects when the input images aren't of the same caliber as the training images.

The focus and scope of this report is to compare and contrast Res-Net-50, VGG-16 and ML-Net in their ability predict salient objects in images that have different types and levels of distortion in order to address the "semantic gap" (Yang, Z. *et al.* 2014) in visual attentive models.

Using the pre-trained models VGG-16, ResNet-50 and ML-Net provided by MIT Saliency Benchmark. The effects of different types and levels of distortion will be assessed by generating saliency maps for Cardiff Universities Distort600 database. Some of the examples of the saliency maps generated by these models can be seen in **Figure 1**, which compares the generated saliency map with ground truth. As seen with the first two images most deep learning models can easily detect the salient object within the image, as the subjects and point of interests are clear. In contrast to the third image, where the saliency maps generated by VGG and ResNet show how inaccurate these predicted maps can be. The VGG predicted

**Figure 1.** Table comparing predicted saliency maps with ground truth images.



map shows a central bias when predicting saliency with images that have no clear point of interest, such is also seen in regards to prediction made by ResNet.

By quantifying the images using Matlab metrics such as CC, Borji, NSS, KL-Div and Judd provided by MIT Saliency. The effects of how these deep learning models perform with varying quality of images can be better analysed through data analysis, to understand to what degree the saliency maps accuracy depends on the input image and its image content.

Saliency maps produced by the deep learning models will be quantified and evaluated to see how they perform with different categories of images, types of distortion and levels of distortion present in the images. The results show that although the metrics used may place some saliency models higher than others, a closer inspection is required in order to understand the evaluation made by the metrics. A deeper analysis by comparing and contrasting the saliency maps and ground truths produced reveals how the saliency can be subjective at times and even us as humans have trouble determining salient objects when there are more than two subjects or when there are no key areas of interest such as in patterns.

# 4. Background

Computer vision has seen several changes throughout the years due to advancements in image processing with deep machine learning, whereby some of the earlier methods of salient object detection have been fundamentally changed. This section aims to contextualise the research of how pre-trained machine learning models perform on images that are not of the same caliber as their initial training images.

## 4.1. Feature Extraction

Earlier models that were used to detect salient objects within images faced a problem where they lacked computational power to process whole images. This lead to the development and use of feature extraction for image processing, to help reduce the amount of information the machines process. Feature extraction is a process of dimensionality reduction which reduces the number of raw image variables into manageable groups, that still accurately, non-redundantly and completely describes the initial set of raw variables. Essentially, reducing data in "higher dimensional space to a lower dimension, whereby it is now in a space with lesser number of dimensions" (Uberoi, n.d.).

Although the feature extraction techniques helps lower the computational requirements for learning models. The saliency maps generated by these models are far below the baseline models that are currently available. An example of the saliency map generate by a classical model can be seen in **Figure 2**.

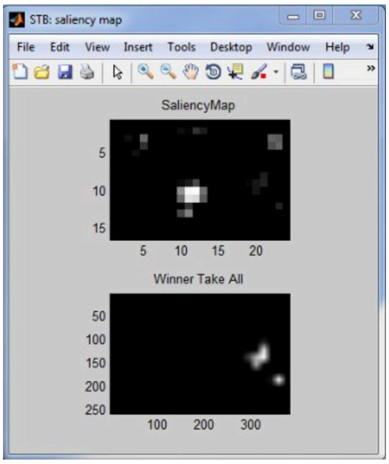


**Figure 2.** Saliency maps generated by a classical model.(Walther and Koch, 2006)

## 4.2. Machine Learning

An application of artificial intelligence (AI) is machine learning, through which systems can learn and improve with experience automatically without being explicitly programmed to do so (Varone, M. *et al.* 2019). By using known examples to learn the relationship between inputs images and ground truths during the training phase; the machine learning algorithm learns to model a function that represents these relationships between the ground truth and input image.

The modelling of the relationship between the given images is achieved through artificial neurones, that function and behave in the same manner as a biological neurone (Uk.mathworks.com, 2019); an innovation inspired by nature, also known as biomimicry. Both biological and artificial neurones possess the same two core capabilities, to integrate signals from different neurones into one single signal and to fire the neurone if the inputs from different neurones cross a certain threshold, also known as the decision boundary. Although this is referred to as the simplest unit of computation in the brain, it is through the interactions of billions of neurones that allows us to perform complex tasks. Similarly, with artificial neurones, it is the interactions between layers of artificial neurones known as the input, hidden and output layer (seen in **Figure 3**) that enables the network to perform complex tasks.

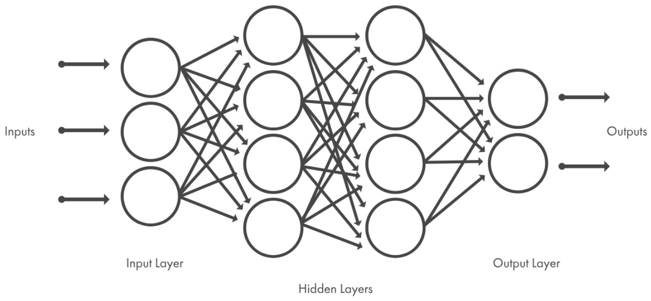


**Figure 3.** Typical neural network architecture. (Uk.mathworks.com, 2019)

In a computational sense the synaptic connections between neurones that are formed over time, is referred to as the weights between each neurone. Artificial neurones learn in the same manner as their biological counterparts which achieves the ability to learn and improve with experience through adjustment of which connections exist and how strong the connections are between neurones.

## 4.1. Deep Learning

Machine learning and deep learning are often used interchangeably, yet they have their own key differences which sets them apart. While machine learning does parse data, learn from it and use it to make informed decisions, it cannot validate and improve on predictions that are wrong. In this case human intervention would be required to address the problem that is resulting in inaccurate predictions. Whereas, deep learning models "can determine on their own if their predictions are accurate or not" (Grossfeld, 2017).

Another key differences in machine learning and deep learning is the number of hidden layers made use by each of the systems. Where a machine learning system consists of one or two hidden layers, most of the deep learning models consist of more than two hidden layers (Gill, 2018). In a Convolutional Neural Network (CNN) which are used to process images, the hidden layer is made up of convolutional layers and other layers such as pooling, fully-connected and normalisation layers.

When machine learning models are trained on unsupervised data, human intervention is required to manually label the data so that the machine learning model can learn from it. In contrast, by using multiple layers, deep learning is able to extract features and label the unsupervised data itself, making them more flexible and less time consuming than machine learning models.

## 4.2.Machine learning models

For the purpose of this research, three different implementations of the neural networks for salient object recognition will be analysed by generating and quantifying the saliency maps. VGG, ResNet and ML-Net have their own methods of analysing salient objects within images. An overview of the models being used can be seen below.

### 4.2.1. VGG-16 & ResNet-50 (SAM-VGG/ResNet)

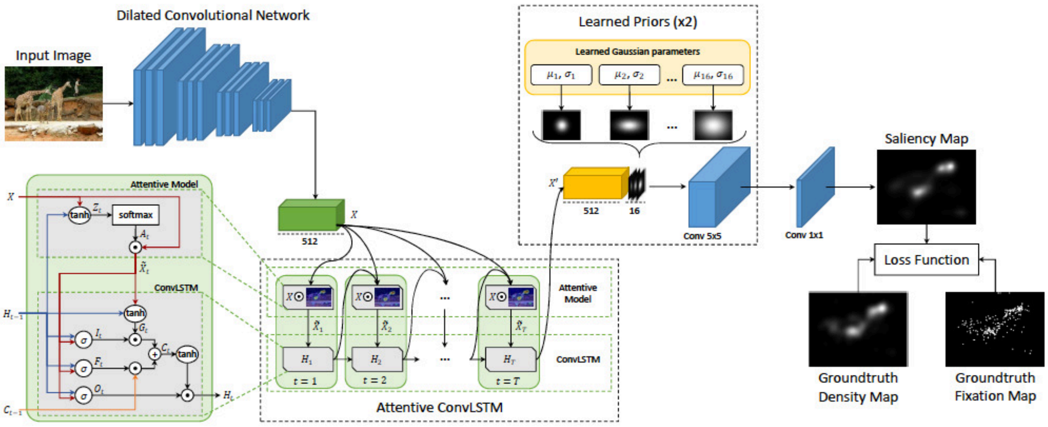


**Figure 4.** Overview of the saliency attentive model (SAM). VGG and ResNet both utilise this architecture which uses a new architecture called Dilated Convolutional Network to compute a set of feature maps. Attentive Convolutional LSTM sequentially enhances saliency features using attentive recurrent mechanism. Learned prior are combined with prediction to model central bias in humans. (Cornia *et al.*, 2018)

### 4.2.2. ML-Net

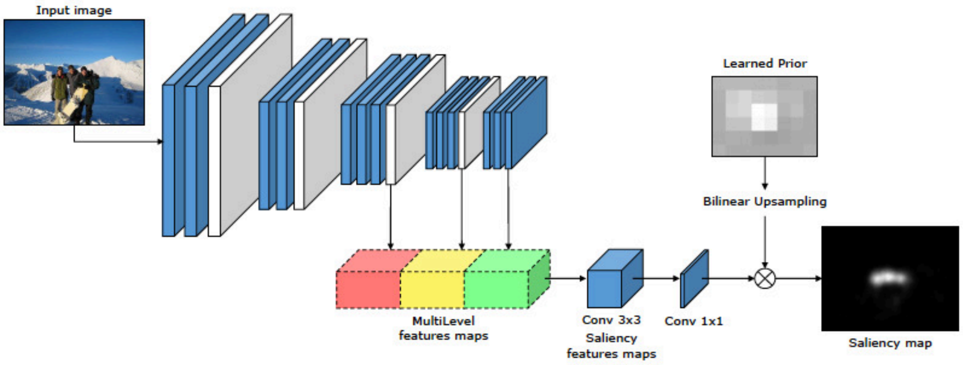


**Figure 5.** Overview of the ML-Net model. Low and high level features are computed using a CNN, feeding the extracted feature maps to an Encoding network. This learns a feature weighting function to generate saliency-specific feature maps. Learned prior is also applied to the predicted saliency map. (Cornia *et al.*, 2016)

## 4.3. Databases

Cardiff University's Distort600 and benchmark released for MIT300 database by MIT Saliency will be used to investigate how deep learning that have been trained on high resolution clean images will perform when the input images are not of the same caliber as the training images. The distortion types and levels present within the Distort600 database can be seen below in **Figure 6**.

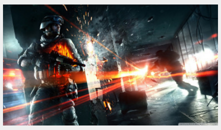
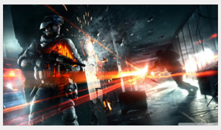
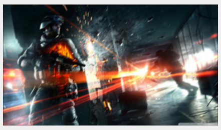
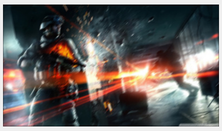
| Distortion type | Original Image | Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| Colour Contrast | | | | |
| JPEG Compression | | | | |
| Motion Blur |  |  |  |  |

**Figure 6.** Table presenting the types and levels of distortion in the Cardiff University, Distort600 database.

The key difference in the two databases being used is that the Distort600 database consists of 600 images in total, of which there are 60 original images that have been distorted by 3 different types such as colour contrast, JPEG compression and motion blur. Each distortion type is also applied with 3 different levels of intensity which consists of level 1, level 2 and level 3.

In contrast the deep learning models that will be tested on the Distort600 database have all been trained and their performance measured and posted online (Bylinskii *et al.*, 2016). By testing the models on a database with different types and levels of distortion, which will allow us to analyse their effects in the accuracy of predictions made by the deep learning models.

## 4.1. Quantifying Saliency Maps

When measuring the similarity between two saliency maps, being able to numerically evaluate their similarity has several advantages when trying to measure their performance. While the field of visual computing is advancing with newer and creative approaches to solving problems, being able to effectively measure the similarity between two saliency maps has been widely debated.

When evaluating saliency maps, MIT Saliency Benchmark has used more than 5 different metrics in order to evaluate the performance of several saliency attentive models. Each metrics  have their own rules and methods of evaluating similarity between the output and ground truth. Metrics such as Kl-div measure divergence between the saliency maps, whereas SIM compares the similarity between the histograms of two saliency maps.

These differences in their approach to measure similarity has lead to more than one metric being used to evaluate these machine learning models. As a specific metric system hasn't been standardised for determining the similarity between two saliency maps. For the purpose of this project, 5 of the relevant metrics and trained deep learning models models provided by MIT Saliency will be used to evaluate their performance on Cardiff University's Distort600 database. These five metrics will be discussed in more detail in the following subsection.

### 4.1.1.Evaluation Metrics Functions

- **AUC_Borji:** This is a version of Area Under ROC curve measure used to benchmark visual attention models by MIT Saliency Benchmark along with other metrics. However, this metric treats the saliency map as a binary classifier to separate false positive(fp) and true positive(tp) by use of various thresholds, where fp rate is the proportion of saliency map values that are above threshold sampled from random pixels and tp is the threshold at fixation locations. This is one of the most commonly-used metric for saliency evaluation. This metric is more sensitive to "high-valued predictions and largely ambivalent of low-valued false positives" (Bylinskii *et al.*, 2017). This metric is also good for detection applications. When quantifying the predicted saliency maps with ground truth, Borji compares the predicted saliency map with the ground truth binary fixation maps.
- **CC:** This is also known as the Pearson's linear coefficient and represents the linear correlation coefficient between two different saliency maps, where CC value of 0 refers to uncorrelated maps. This metric treats false positives and false negatives symmetrically. CC metric compares the predicted saliency map and the ground truth saliency map.
- **AUC_Judd:** Similar to AUC_Borji, the saliency map is treated as a binary classifier, with true positives (tp) rate proportion of saliency map being above threshold at fixation locations and false positive (fp) rate being the proportion of saliency map values above threshold at non-fixated locations. The implementation used and provided by MIT Saliency has threshold values sampled at fixed step size. Judd metric compares the predicted saliency with ground truth binary fixation map.
- **Kl-div:** This metric measures the divergence between two different saliency maps viewing them as distributions. Being a non-symmetric measure of information lost, it highly penalises mis-detection when saliency map is used to predict the fixation maps. Kl-div compares the predicted saliency maps with ground through saliency maps.

- **NSS:** Abbreviated for normalised scan path saliency which is measured as the mean value of the normalised saliency map at fixation locations. It is a "discrete approximation of CC that is additionally parameter-free" (Bylinskii *et al.*, 2017) which operates on fixation maps. This metric compares the predicted saliency map with the ground truth fixation maps and is recommended for saliency evaluation.
- **SIM:** Also known as the histogram intersection, it is a fast and easy method of evaluating similarity between two different saliency maps. By measuring the distribution of histograms produced by the two saliency maps, it is able to evaluate the predicted saliency map on its accuracy. This metric compares the predicted saliency map with the ground truth saliency map and is more sensitive to false negatives than false positives. However, this metric does assume that the inputs are valid distributions.

# 5. Approach & Implementation

When carrying out the investigation of how distortion effects the predictions made by deep learning models that have been trained on clean, high resolution images; the initial approach to investigating their effects was by implementing the three neural networks. Although the code and weights for the neural networks were provided through MIT Saliency benchmark. It is not guaranteed to successfully run without complications as libraries and packages used might become outdated.

## 5.1. Setting up environment

The environments for the networks were set up using virtual environments, this allowed for multiple environments to exist with their own packages and dependencies. Anaconda was used to create these virtual env for the networks. Although the networks have similar requirements, the requirement for ML-Net differs slightly from both VGG and ResNet. To ensure the testing was fair between the different neural networks, as a part of the approach, an environment was made for each of the networks.

Relevant packages such as Theano 0.9.0, OpenCV 3.0.0 and Keras 1.1.0 using Theano as backend were installed for each of the environments. In order to use Theano as backend the hidden file called "keras.json" needs to be accessed and made sure to have "image_dim_ordering": "th" and "backend": "Theano".

### 5.1.1. Packages compatibility issues

Initial issues were seen when the python version used wasn't stated. The information about which python is used in creating these environments also weren't available on Github. However, this compatibility issue was addressed by trying to implement the network using different python versions. Python 3.7 was ruled off through some research, which lead to the discovery that some of the Keras version required wasn't compatible with the latest python version. Installing python 3.7 lead to a higher version of Keras being installed, where the variable names had been changed from "initializations" to "initializers" (Keras.io, n.d.).

Moreover, the packages also had to be installed in a specific order, which wasn't stated in the requirements, this lead to a subtle change in the versions of Theano being used. If the correct version of Theano was installed in the beginning, this version would be overwritten by the Theano installed by Keras. This lead to several errors and the network not functioning, yet it was solved by checking whether the correct packages were installed.

The problem in setting up the environments was also seen as the correct version of OpenCV couldn't be installed with the current packages. OpenCV 3.0.0 was also only available with linux machines. However, this lead to using the version 3.1.0 which is available for both Mac and Windows platform through the condo navigator. The commands *conda install -c menpo opencv3* was used to install this specific version of OpenCV 3.1.0. As this was not the version of OpenCV stated on the requirements, saliency maps were generated to see if the difference in version would still produce saliency maps without causing errors during runtime.

### 5.1.2. Pre-trained weights

When training a neural network, weights between each of the different neurones in the hidden layers are recorded and packaged into containers or pickle files so that they can be easily transferred and used to make predictions, without having to train the model each time.

MIT saliency provides the pre-trained weights generated from training their neural network with their MIT300 database, consisting of clean, high-resolution images as a pickle file with the extension .pkl. The pickle file needed to be opened and loaded onto the neural network, opening the pickle file resulted in failure. However, through closer inspection of the code they had provided for the neural networks, they had built-in functions to handle the opening and loading of the weights from the pickle files. The necessary weights for each of the networks were then downloaded and moved to the same directory as the neural network.

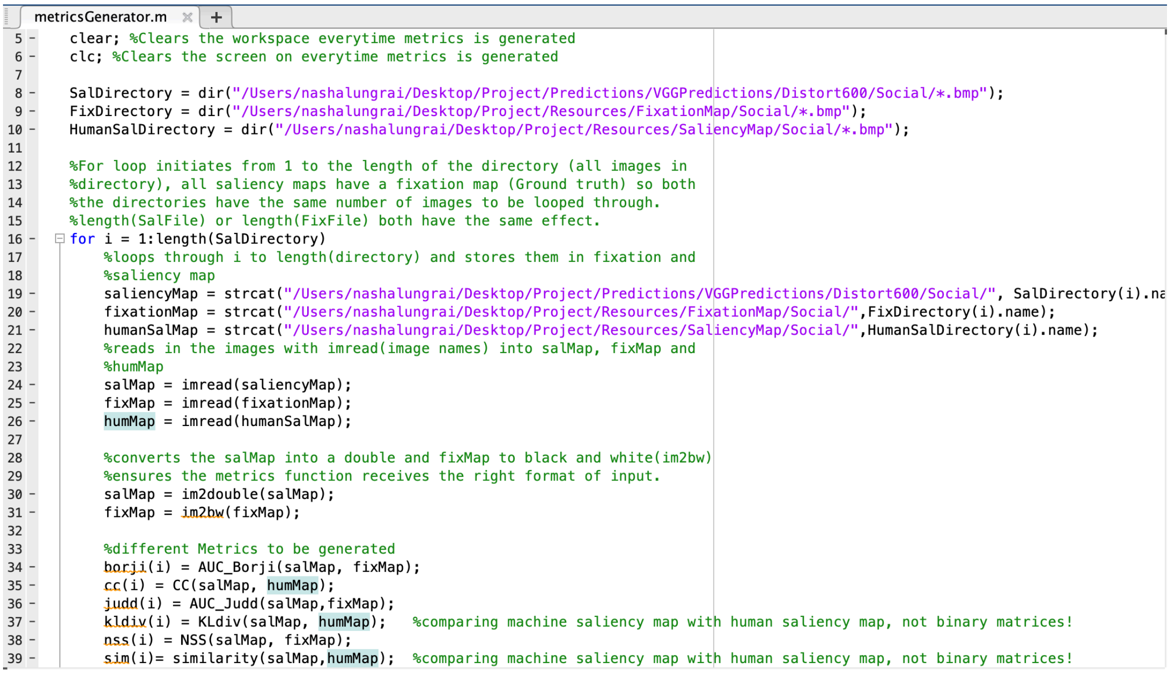
### 5.1.3. Generating Saliency Maps

Computing the saliency map with the pre-trained model was done through the command provided in the documentation of the networks. The command being, "*python main.py test path/to/images/folder/*" where path/to/images/folder refers to the path of folder containing images that need saliency maps to be computed for.

While generating the saliency maps for the database, this was done through batch processing. Since the database was provided being separated into 10 different categories of images, each category containing 60 images. Each category of images were calculated one at a time as more images required longer time for computing the saliency maps, which could cause the laptop to heat up. Hence, computing the saliency map was done in batches, depending on the categories.

## 5.2. Generating data in MATLAB

Matlab metrics code was provided by MIT Saliency for the 5 metrics that will be used to quantify the similarity between two saliency maps. The same metrics code was used to calculate the similarity as it would uphold the validity of the results and removes any uncertainty as the same metrics are being used and compared. Although the metrics code was provided, the code currently only computes the similarity between two saliency maps. However, in order to compute the five different metrics of all images within a directory a custom loop had to be created in Matlab.



```
clear; %Clears the workspace everytime metrics is generated
clc; %Clears the screen on everytime metrics is generated

SalDirectory = dir("/Users/nashalungrai/Desktop/Project/Predictions/VGGPredictions/Distort600/Social/*.bmp");
FixDirectory = dir("/Users/nashalungrai/Desktop/Project/Resources/FixationMap/Social/*.bmp");
HumanSalDirectory = dir("/Users/nashalungrai/Desktop/Project/Resources/SaliencyMap/Social/*.bmp");

%For loop initiates from 1 to the length of the directory (all images in
%directory), all saliency maps have a fixation map (Ground truth) so both
%the directories have the same number of images to be looped through.
%length(SalFile) or length(FixFile) both have the same effect.
for i = 1:length(SalDirectory)
    %loops through i to length(directory) and stores them in fixation and
    %saliency map
    saliencyMap = strcat("/Users/nashalungrai/Desktop/Project/Predictions/VGGPredictions/Distort600/Social/", SalDirectory(i).na
    fixationMap = strcat("/Users/nashalungrai/Desktop/Project/Resources/FixationMap/Social/",FixDirectory(i).name);
    humanSalMap = strcat("/Users/nashalungrai/Desktop/Project/Resources/SaliencyMap/Social/",HumanSalDirectory(i).name);
    %reads in the images with imread(image names) into salMap, fixMap and
    %humMap
    salMap = imread(saliencyMap);
    fixMap = imread(fixationMap);
    humMap = imread(humanSalMap);

    %converts the salMap into a double and fixMap to black and white(im2bw)
    %ensures the metrics function receives the right format of input.
    salMap = im2double(salMap);
    fixMap = im2bw(fixMap);

    %different Metrics to be generated
    borji(i) = AUC_Borji(salMap, fixMap);
    cc(i) = CC(salMap, humMap);
    judd(i) = AUC_Judd(salMap,fixMap);
    kldiv(i) = KLdiv(salMap, humMap);   %comparing machine saliency map with human saliency map, not binary matrices!
    nss(i) = NSS(salMap, fixMap);
    sim(i)= similarity(salMap,humMap);  %comparing machine saliency map with human saliency map, not binary matrices!
```

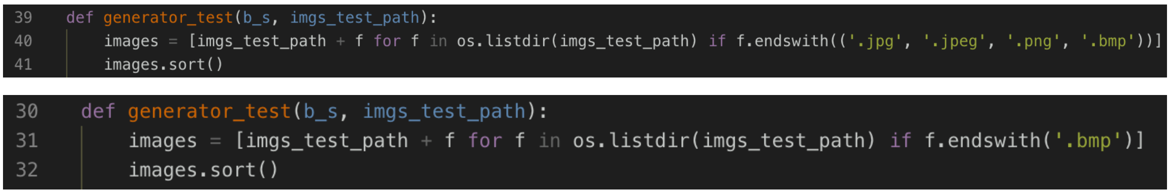
**Screenshot 1.** Code written in Matlab to calculate similarity of all images within a directory.

As seen in **Screenshot 1**, a custom loop was made in order to navigate to the correct directories for the fixation maps, saliency maps and predicted saliency maps. Where salMap refers to the model generated saliency maps, fixMap are the binary fixation maps (Ground truth) and humMap are the ground truth saliency maps.

### 5.2.1. File Extension Issues

When generating the similarity metrics, one of the issues was caused due to the file extensions used for predicted saliency maps and ground truth maps were different. Hence, all of the file extension for the database and predicted saliency maps were change to the bitmap image file extension (.bmp).

Similar issue arises when computing the saliency maps using neural networks. The code provided for ResNet and VGG were easily modifiable to allow it to compute saliency maps for images with different file extension at once. However, ML-Net could only compute saliency maps for one specific file extension at a time, saliency maps



```
def generator_test(b_s, imgs_test_path):
    images = [imgs_test_path + f for f in os.listdir(imgs_test_path) if f.endswith(('.jpg', '.jpeg', '.png', '.bmp'))]
    images.sort()
```

```
def generator_test(b_s, imgs_test_path):
    images = [imgs_test_path + f for f in os.listdir(imgs_test_path) if f.endswith('.bmp')]
    images.sort()
```

**Screenshot 2.** The first screenshot shows the code showing different file extensions accepted by VGG & ResNet at once. Whereas, the second screenshot shows that ML-Net can only run with one file extension at a time.

were generated for the first file extension listed in the code, but all the other extensions were ignored. Overcoming this issue was done by changing the file extension in the code and running it for .png, .jpg and .bmp separately. This can be seen in **Screenshot 2**, where VGG and ResNet have can generate saliency maps for different file extensions at once, whereas ML-Net can only generate saliency maps for one file extension every time it runs.

## 5.1. Restructuring Databases

Initially the Cardiff University's Distort600 database was split into 10 different categories. However, when generating the metrics for different distortion types and levels, the initial structure of the database doesn't allow for easy metrics generation. In order to carry out this generation of metrics, two new databases were made for the predicted saliency maps, fixation maps and ground truth saliency maps. One of the database organised by different levels of distortion and the other was by distortion types. The analysis and discussions of the metrics generated from these databases will be in the following section.

# 6. Data Analysis & Discussions

## 6.1. Overall Analysis

The similarity metrics generated from Matlab were recorded and processed using Microsoft Excel and Numbers. **Figure 7** shows the overall performance of the pre-trained models on the MIT300 and Distort600 Databases.

**A** shows the overall performance measured and published by MIT. This clearly shows the performance of the pre-trained models being higher on the MIT300 database then the performance measured on Distort600 database, which can be seen in **B**. The similarity performance measured by the metrics Borji, CC, Judd and SIM are seen to be more consistent across all of the three pre-trained models when tested on the MIT300 database. However, there are inconsistencies in the results seen when tested on the Distort600 database. The metrics fluctuate between the three models which would be due to the presence of distortion types and levels of distortion present within Distort600. Hence, the performance of these pre-trained models aren't as stable in **B** as they are in **A**.

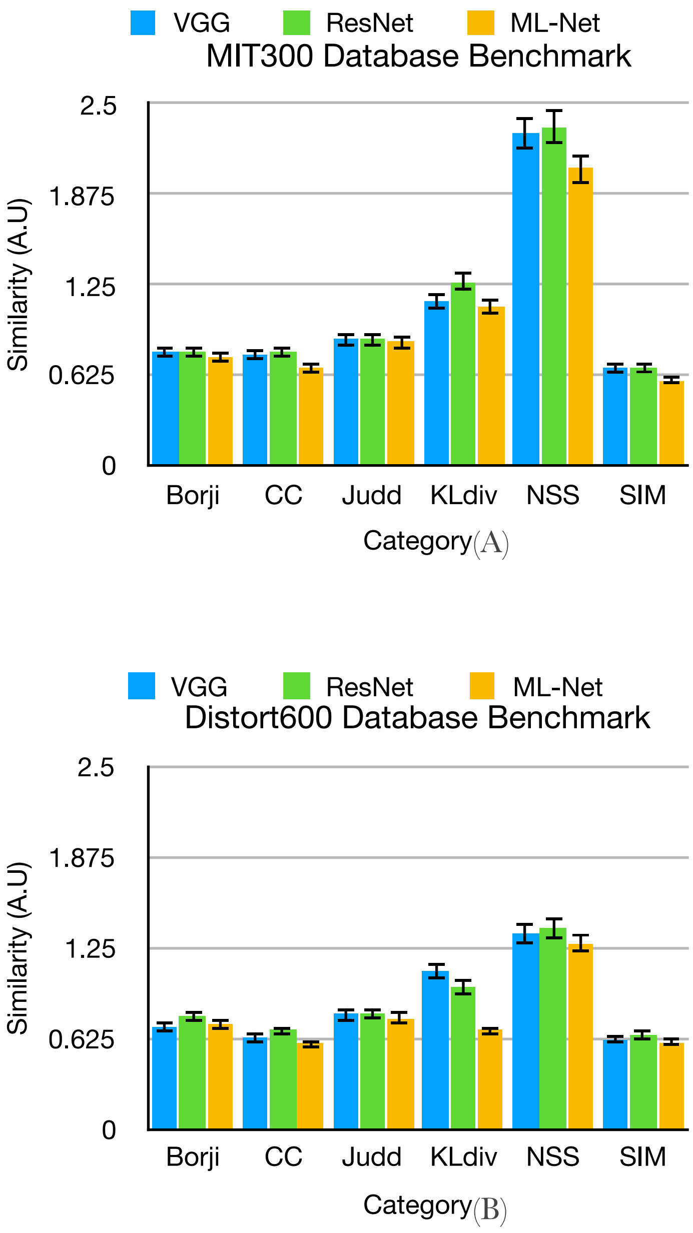


**Figure 7.** Graphs show the overall performance of pre-trained models measured by 5 metrics on Distort600 and MIT300 database.

The NSS and KL-Div values show a significant decrease in performance on Distort600 than on MIT300. Whereas, the other metrics such as Borji, CC, Judd and SIM only show a slight decrease in performance. This would be caused by the difference in the method of evaluating similarity between two saliency maps. KL-Div compares the predicted and ground truth saliency maps and highly penalises any mis-detections in the saliency map. Since the models haven't been trained on distorted images, it could lead to inaccuracies in the predicted map, leading to the significant performance difference of models. Where KL-Div measures

ResNet as being the most accurate at predicting salient features with no overlap between the 95% confidence error bars, followed by VGG and ML-Net which shows an overlap in their performance. Whereas, their performance on Distort600 shows VGG having better predictions, followed by ResNet and ML-Net. From this inspection of the graphs it could be said that although the performance does decrease when a pre-trained model is given distorted images, VGG is still better at handling these distortions and correctly detecting the salient objects within images.

Since NSS, is recommended for the saliency evaluation, analysis of the histograms produced in **Figure 8** for the NSS metric with different models reveals the distribution of the similarity evaluation for entire Distort600. Where the histogram for VGG shows a left skewered graph, showing that most of the evaluations of the saliency maps produced are on the lower end of the distribution. This would be due to the evaluation by NSS of the distorted images, which would yield a lower evaluation of the similarity between predicted saliency map and ground truth. Whereas, ResNet shows a slight left skewered but not to the extent of VGG, as more predictions made by ResNet are closer to the mean of 1.39, while the mean of VGG

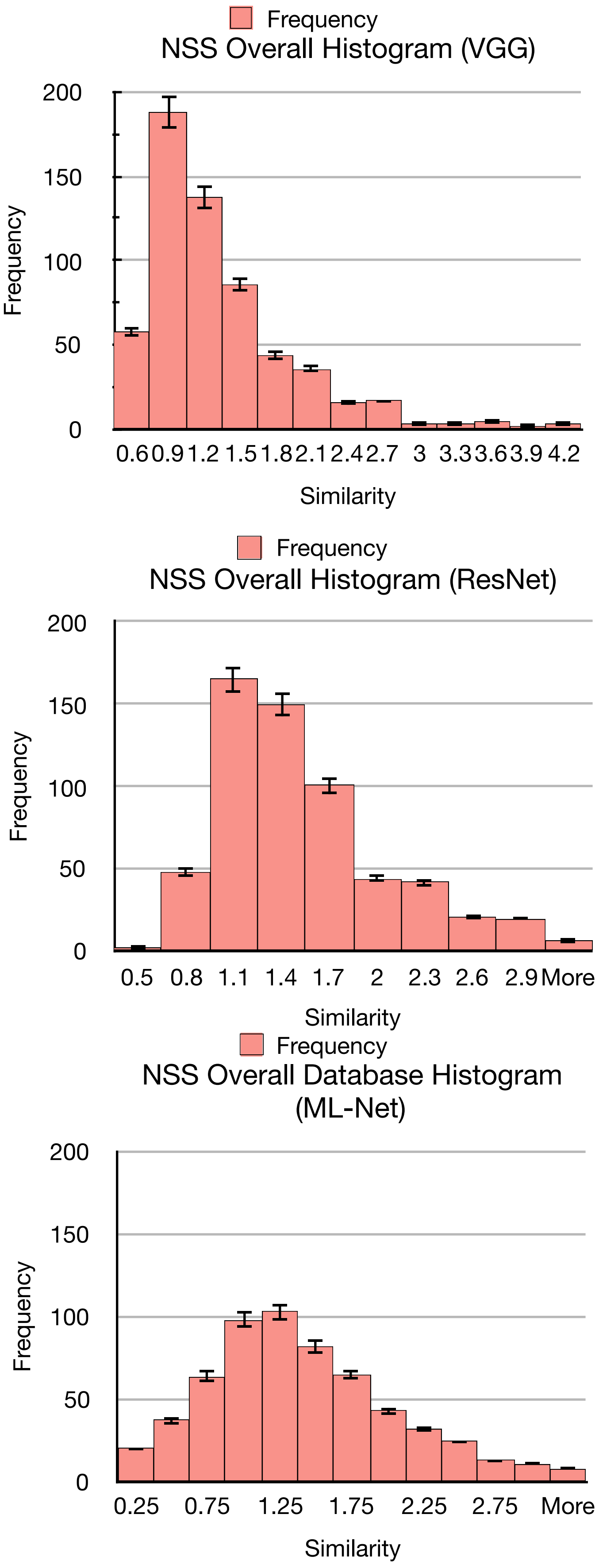


**Figure 8.** Overall histogram of the NSS evaluation generated for pre-trained models.

(1.35) is further away from the higher frequency of evaluation. In the case of ML-Net a normal distribution curve is seen with the mean of 1.28. This shows that even though VGG and ResNet models yield more inconsistent results due to the presence of distorted images, while ML-Net performs better overall in terms of the data yielded from the NSS metric.

## 6.2. Categories Analysis

A closer inspection of how the metrics evaluate the performance of VGG on the different categories reveals the characteristics of the different metrics. As seen in **Figure 9**, the performance of the metrics shows that majority of the metrics are consistent throughout the different categories. With most of the metrics showing an increase in performance in regards to categories such as portrait. Since, portrait images have a clear focus of subject and background, detection of these types of features is easier than that of detecting salient objects within the patterns category.

Further analysis of the graph also reveals that while majority of the metrics are consistent in their performance evaluation, both NSS and KL-Div show more fluctuations between the categories than majority of the metrics. This could be due to their method of highly penalising mis-detections in the saliency maps and the methods they use to evaluate the saliency maps.

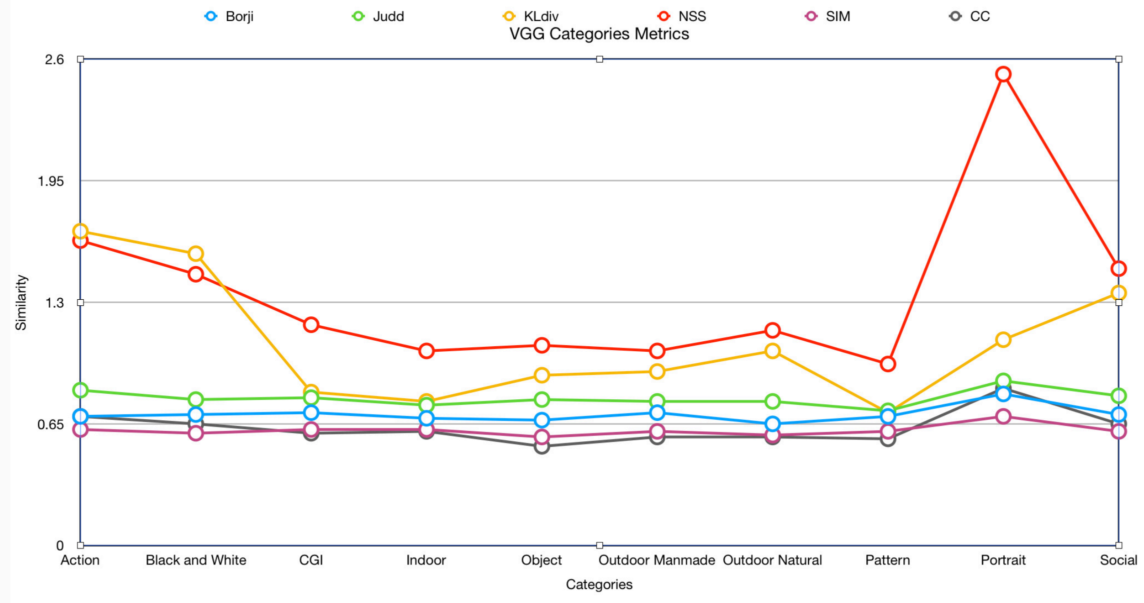


**Figure 9.** Graph shows the performance of VGG within different categories.

Although majority of the metrics are consistent and show slight fluctuations, analysis of how the pre-trained model performs with the Distort600 can be better analysed with the use of NSS as it is recommended for saliency evaluation. The NSS evaluation also shows distinct decreases and increases in regards to the categories.

The NSS metric shows a clear decrease in performance in categories such as indoor, object, outdoor manmade and outdoor natural. With the lowest performing category being pattern. When viewing images in the pattern category, even humans cannot come to a conclusion as to what the salient objects might be, which may cause the machine to predict these images with higher inaccuracies as there is no clear

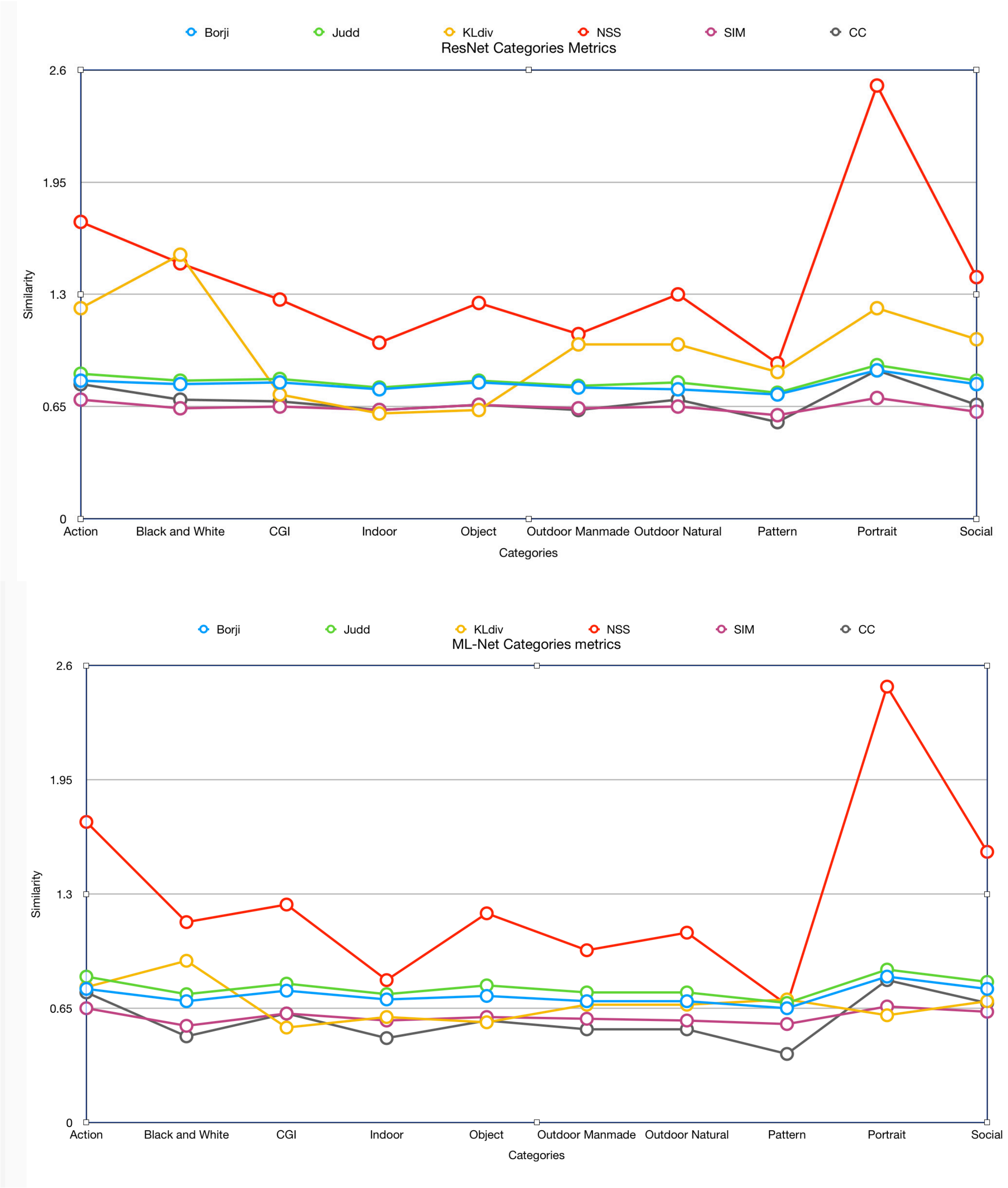


**Figure 10.** Graphs show the performance of ResNet & ML-Net within different categories.

indication of the subject and background. As seen in **Figure 10** both ResNet and ML-Net also show pattern as being their lowest performing category for predictions. On the other hand, portrait, social and action are rated as the highest performing categories  amongst all of the models by NSS.

## 6.3. Distortion Type Analysis

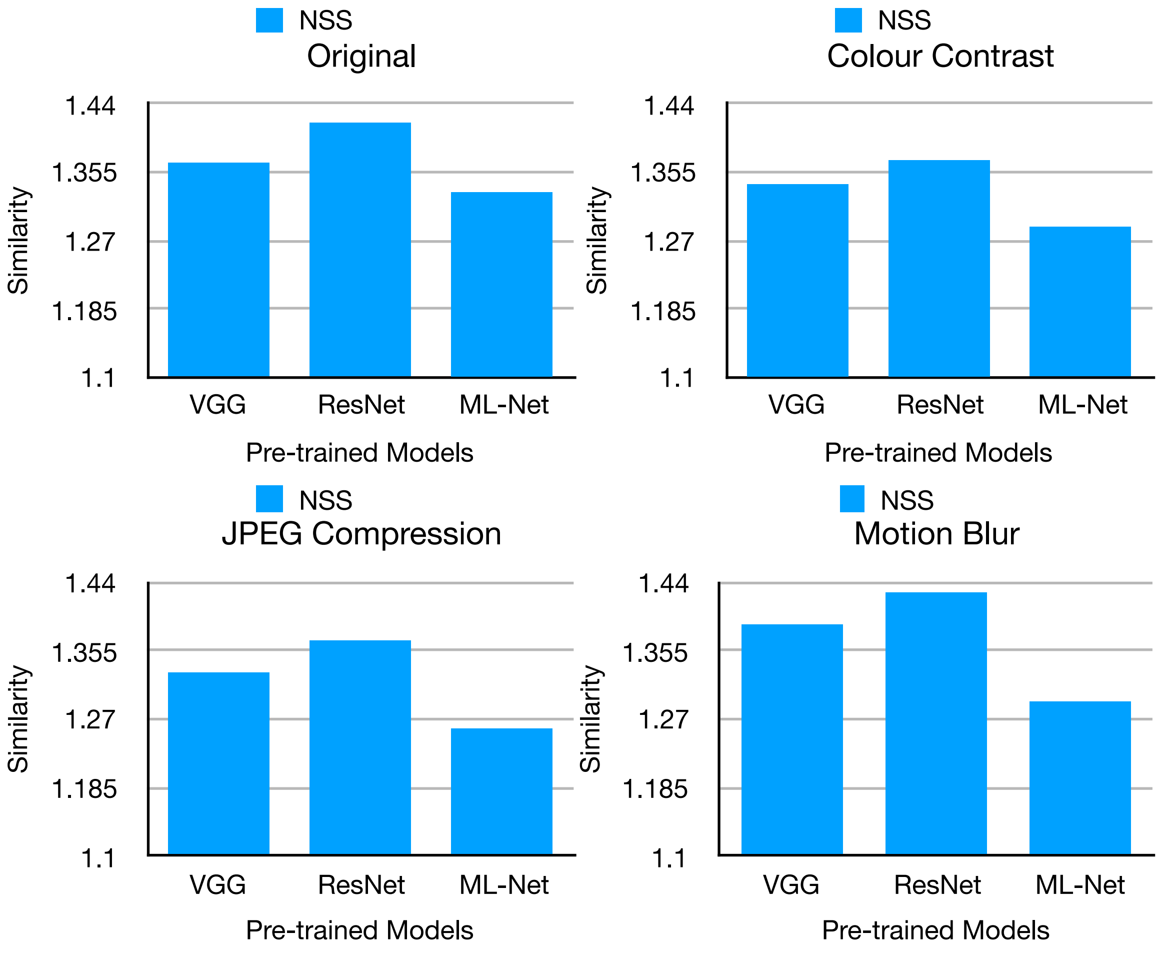


**Figure 11.** Graphs showing the performance of pre-trained models on different types of distortion, measured by NSS.

Distort600 contains images with different types of distortion, which could effect the accuracy when predicting saliency maps. As seen in **Figure 11**, the bar graphs show the effect on the performance of the three pre-trained models when different types of distortion are introduced. Both VGG and ResNet show a trend of decreasing performance when colour contrast and JPEG compression but also shows better performance than with original image, when dealing with images that contain motion blur. Whereas, ML-Net shows better performance with original images than with images that contain any form of distortion.

Lower performance seen in colour contrast and JPEG compression may be due to the type of distortion that is applied onto the photos. As seen in **Figure 6**, section 4.3, colour contrast and JPEG compression both effect the colour composition of the photo, where by applying it in various levels, results in over exposure of the background with colour contrast and segmented change in colour seen with the application of JPEG compression. Both of which may result in model under-fitting when it comes to generating saliency maps with the pre-trained models as they haven't been trained on images that contain such distortion. In addition, as the colour composition of the background is changed, the ground truth also changes, as these distortions may cause the background to become the more salient part of the image, drawing some human attention towards the area.

In **Figure 12** the comparisons made between the predicted saliency maps and the ground truth could give us more information about the why motion blur yields higher performance with VGG and ResNet models. When comparing motion blur distortion with ground truth images, VGG produces saliency maps that has two points of interest, seen as the dense white areas in the map. Whereas, ResNet produces a saliency map with three points of interest, that form a bulbous clusters that are connected. In comparison to the saliency map produced by ML-Net contains two points of interest with other smaller points of interest, which is much more similar to the ground truth than the saliency maps for VGG and ResNet.

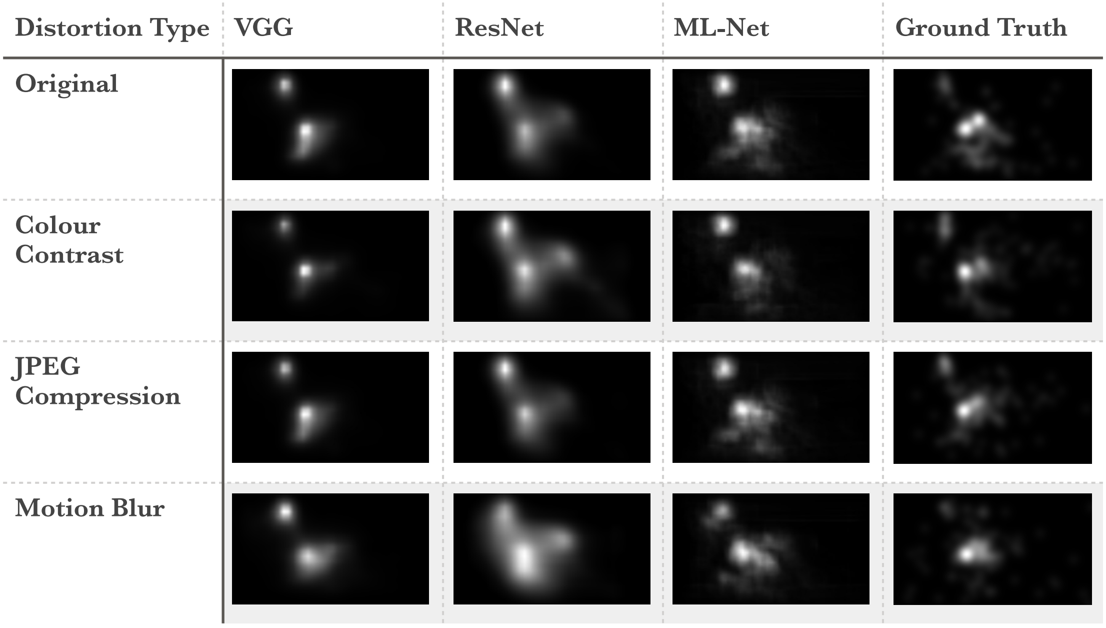


**Figure 12.** Table comparing the different saliency maps generated by pre-trained models for level 3 distortion, along with the ground truth maps.

Since ResNet performs better than other models with every type of distortion present in Distort600, this could be due to the nature of the saliency maps being produced. By having most of the predictions in a bulbous cluster, larger number of fixation points and areas of interest seen in ground truth saliency maps would be covered by the generated maps, resulting in higher performance with certain metrics.

VGG has distinct areas of interest, which may lead to smaller areas of interest being ignored by the model, as seen in colour contrast and JPEG Compression. However, ML-Net produces saliency maps that take into account the smaller points of interest, as the ground truth for motion blur shows one main region of interest surrounded by smaller areas of interest. When comparing the ground truth to the ML-net saliency map, it is more likely to cover the same salient objects within the image.

Additionally, when comparing the saliency maps in the **Figure 12**, the saliency maps generated by ML-Net are more similar to what the ground truth maps than the maps produced by ResNet and VGG models.

## 6.1.Distortion Level Analysis

Distort600 also contains different levels of distortion, the performance of deep learning models on their accuracy to predict the salient objects within the levels of distortion will be discussed. The line graph shown in **Figure 13**, shows the performance of these pre-trained models on different levels of distortion within the images. Both ResNet and VGG show higher performance when dealing with original images and show a sharp decline in performance when dealing with the first level of distortion. VGG shows an increase in performance in level 2 distortion but a decline in level 3 where its predictions are closer to that of level 1. Similarly the accuracy of prediction for ResNet is seen to have a slow and steady increase in its performance at higher levels. The performance of ML-Net contrasts the trends followed by VGG and ResNet, where it performs worse on original images but its performance continues to

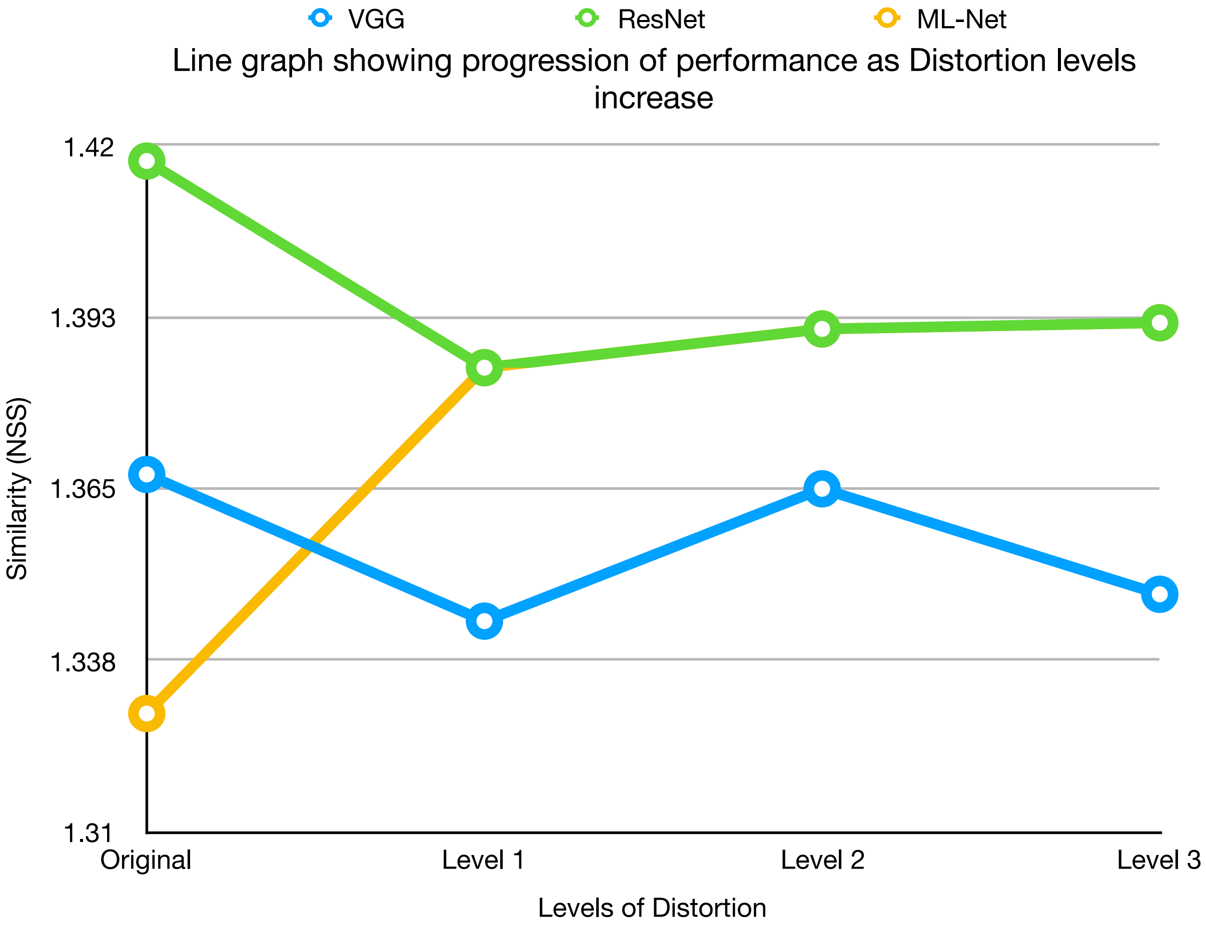


**Figure 13.** Line graph shows how the pre-trained models perform at different levels of distortion.

increase with higher levels of distortion, following the same pattern as ResNet after level 1.

The decrease in performance at level 1, followed by an increase at level 2 could be due to the amount of distortion being used. When analysing the images for salient objects the boundaries between distorted and non-distorted images may be confused by the weights used by pre-trained models. This causes a sharp increase is seen at level 2, as it is able to look past the distortion to produce a higher metric evaluation. When looking at distorted images humans also tend to look past some of the distortion and still see the salient object underneath the distortion.

However, since the pre-trained models have been trained on clean, high-resolution images, they aren't able to predict the salient features as accurately as they can with distorted images. Seen in **Figure 14**, the table compares the saliency maps produced by VGG, ResNet and ML-Net with the ground truth. VGG correctly predicts the most salient object within the photo (the brightest point in the saliency map), while the smaller salient features are not detected. ResNet follows the previously seen bulbous cluster when identifying salient objects, which leads it to cover the salient objects within the ground truth, ensuring that its evaluation of similarity is better. Whereas, the saliency map produced by ML-Net more accurately addresses the smaller salient points within the image, which results in it performing better than other models when it comes it images with distortion.

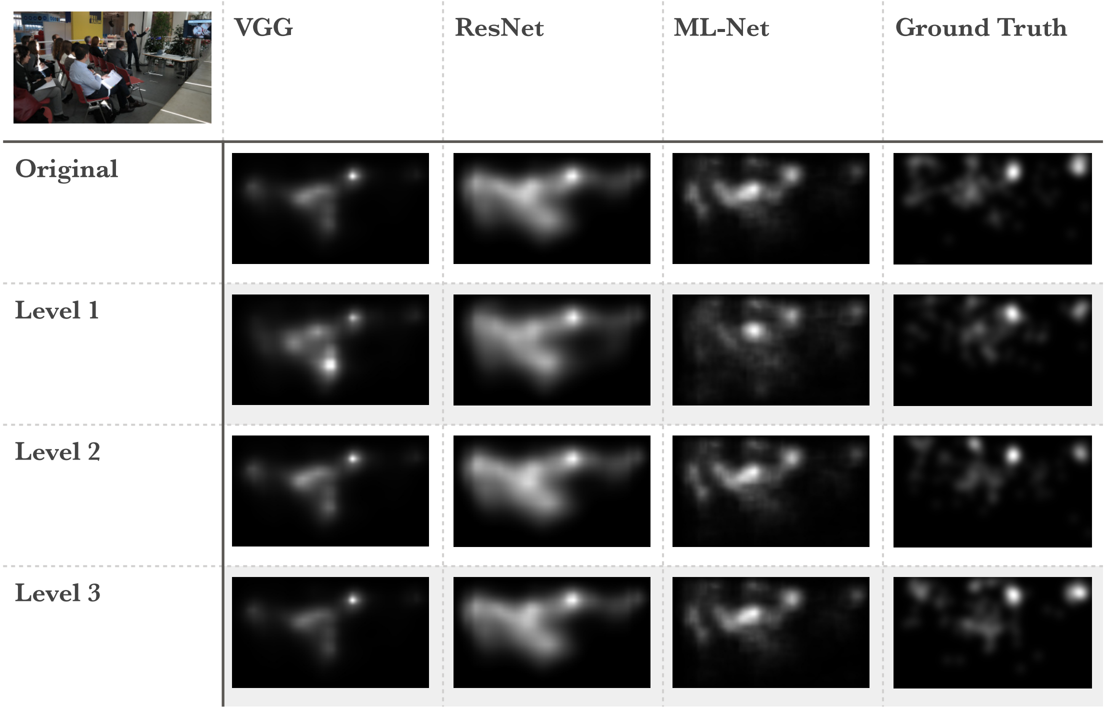


**Figure 14.** Table compares the saliency map from different pre-trained models with ground truth images.

When considering the input image seen in the top right corner of **Figure 14**, it shows that the salient objects recognised by both VGG and ResNet include the faces and humans present in the crowd. Since, VGG and ResNet are neural networks used for object detection, the implementation of it may still have a bias for detecting faces and objects as salient objects. Yet, the ground truth reveals that when the picture was displayed for data collection, humans focused more on the background objects such as the texts on boards, face of the presenter and the back of peoples' heads. This is much closer to the prediction made by ML-Net as it tries to predict the smaller salient objects and doesn't seem to have the same bias as VGG and ResNet models.

Another feature of the saliency maps from VGG and ResNet show that these deep learning models lack the ability to detect depth within the image. Both the models' saliency maps show the objects being detected as salient are the objects that are the closest to the camera in the image. This results in the saliency maps being generated by VGG and ResNet to the crowd being the most salient object within the image.

When considering images with more than one subject, it is also important to remember that even amongst humans there may be disagreements as to where they look in specific images. Since there are more than one salient object within the image seen above, the areas where humans focus on can be dependent on the person and quite subjective.

# 7. Conclusion

From analysis of the different deep learning models for salient object recognition, the results show that there is an overall decrease in performance when the pre-trained models are used to compute the saliency maps for Distort600. Although, the general decrease may be due to the categories of images such as pattern, social and outdoor natural not being used to train the deep learning model. Hence, the predictions made for these categories yield a lower performance score using the similarity metrics, which decreases the overall performance of the models on Distort600 database.

Further analysis of the different distortion types also revealed that even though VGG and ResNet are seen to perform better when looking at the evaluation from different metrics. The saliency maps themselves show, central biases in their predictions, which could be due to the learned prior being combined with predictions to model humans central bias.

ML-Net has an increase in its performance when the levels of distortion increases, which due to the smaller salient features being predicted by the model, unlike, ResNet and VGG which form more of a bulbous cluster as a saliency map.

In the future, some of the limitations could be addressed by making improvements such as training the neural networks with distorted and clean images in order to improve its performance when detecting salient objects within distorted images. By training the images with distorted images, the models are more likely to perform better with distorted images as the weights used by the models are adjusted for the distortion as well.

Improvements could also be made by making use of Spatial Pyramid Pooling (SPP), which is can be located between the last convolutional layer and the fully-connected layer. Since the fully-connected layer requires a fixed sized input image, the SPP layer pools the features extracted by the convolutional layer and returns a fixed length output. This provides 24 - 102 times faster image processing than R-CNN and slightly better than a fine-tuned R-CNN (He *et al.*, 2014). This could lead to faster prediction generation time for images when using the models. Although this hasn't been considered in context for CNNs it also has other benefits such as pooling extracted features at different scales and its use of multi-level spatial bins which robust to object deformation.

# 8. References

- Bylinskii, Z., Judd, T., Oliva, A., Torralba, A. and Durand, F. (2017). What Do Different Evaluation Metrics Tell Us About Saliency Models?. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [online] 41(3), pp.740-757. Available at: https://arxiv.org/pdf/1604.03605.pdf [Accessed 1 Apr. 2019].
- Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A. and Torralba, A. (2016). *MIT Saliency Benchmark*. [online] Saliency.mit.edu. Available at: http://saliency.mit.edu/results_mit300.html [Accessed 4 Feb. 2019].
- Cornia, M., Baraldi, L., Serra, G. and Cucchiara, R., 2016, December. A deep multi-level network for saliency prediction. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (pp. 3488-3493). IEEE.
- Cornia, M., Baraldi, L., Serra, G. and Cucchiara, R., 2018. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, *27*(10), pp. 5142-5154.
- Grossfeld, B. (2017). *A simple way to understand machine learning vs deep learning - Zendesk*. [online] Zendesk. Available at: https://www.zendesk.com/blog/machine-learning-and-deep-learning/ [Accessed 9 Apr. 2019].
- Gill, J. (2018). *Automatic Log Analytics using Deep learning and AI - XenonStack*. [online] XenonStack. Available at: https://www.xenonstack.com/blog/log-analytics-deep-machine-learning/ [Accessed 11 Apr. 2019].
- He, K., Zhang, X., Ren, S. and Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), pp.1904-1916.
- Itti, L. (2007). *Visual salience*. [online] scholarpedia.org. Available at: http://www.scholarpedia.org [Accessed 2 Apr. 2019].
- Keras.io. (n.d.). *Initializers - Keras Documentation*. [online] Available at: https://Keras.io/initializers/ [Accessed 10 Apr. 2019].
- Uberoi, A. (n.d.). *Introduction to Dimensionality Reduction - GeeksforGeeks*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/dimensionality-reduction/ [Accessed 8 Apr. 2019].
- Uk.mathworks.com. (2019). *What Is a Neural Network?*. [online] Available at: https://uk.mathworks.com/discovery/neural-network.html [Accessed 8 Apr. 2019].
- Varone, M., Mayer, D. and Melegari, A. (2019). *What is Machine Learning? A definition - Expert System*. [online] Expertsystem.com. Available at: https://www.expertsystem.com/machine-learning-definition/ [Accessed 4 Apr. 2019].
- Walther, D. and Koch, C., 2006. Modeling attention to salient proto-objects. *Neural networks*, *19*(9), pp.1395-1407.

- Yang, Z., Koch, C. and Zhao, Q. (2014). *Neural computation, neural devices, and neural prosthesis*. New York: Springer Science+Business Media New York, pp.338-339.

# School of Computer Science and Informatics

# Coursework Submission Cover Sheet

***Please use Adobe Reader to complete this form. Other applications may cause incompatibility issues.***

| | |
|---|---|
| Student Number | C1630345 |
| Module Code | CM3203 |
| Submission date | 10/05/19 |
| Hours spent on this exercise | 60+ |
| Special Provision | [redacted] |

(Please place an x is the box above if you have provided appropriate evidence of need to the Disability & Dyslexia Service and have requested this adjustment).

## Group Submission

For group submissions, *each member of the group must submit a copy of the coversheet.* Please include the student number of the group member tasked with submitting the assignment.

| | |
|---|---|
| Student number of submitting group member | [redacted] |

***By submitting this cover sheet you are confirming that the submission has been checked, and that the submitted files are final and complete.***

## Declaration

***By submitting this cover sheet you are accepting the terms of the following declaration.***

I hereby declare that the attached submission (or my contribution to it in the case of group submissions) is all my own work, that it has not previously been submitted for assessment and that I have not knowingly allowed it to be copied by another student. I understand that deceiving or attempting to deceive examiners by passing off the work of another writer, as one's own is plagiarism. I also understand that plagiarising another's work or knowingly allowing another student to plagiarise from my work is against the University regulations and that doing so will result in loss of marks and possible disciplinary proceedings.