Bachelor Thesis

Analysis and measurement of visuospatial complexity

Alissar Al Saleh
Computer Science

Studies from the School of Science and Technology at Örebro University
Örebro 2023

Studies from the School of Science and Technology at Örebro University

Alissar Al Saleh

Analysis and measurement of visuospatial complexity

Abstract

The thesis performs an analysis on visuospatial complexity of dynamic scenes, and more specifically driving scenes in the propose of gaining a knowledge on human visual perception of the visual information present in a typical driving scene. The analysis and measurement of visual complexity is performed by utilizing two different measure models for measuring visual clutter, Feature congestion clutter measure [1] and Subband entropy clutter measure[1] introduced by Rosenholtz, a cognitive science and research. The thesis represent the performance of the computational models on a data set consisting of six episodes that simulate driving scenes with different settings and combination of visual features. The results of evaluating the measure models are used to introduce a formula for measuring visual complexity of annotated images by extracting valuable information from the annotated data set using Scalabel[2], an annotation web- based open source tool.

Keywords

Computer Science,Artificial Intelligence, Visuospatial Complexity, Feature Congestion, Subband Entropy, Visual Clutter, Image annotation.

Contents

Chapter 1
Introduction

In recent years, the development of AI systems has rapidly increased making these systems gain a great attention and become widely desired and used in different fields. Their impact on humans have also become an interesting subject for investigation in different fields. One area of particular interest is the development of autonomous cars,which uses multi-modal interaction to navigate and interact with their surroundings. However,researches on such technologies focuses mostly on the importance of functionality, utility, and usability of these systems, but rarely addresses the challenge in their design and capability of simulating human perception of complex and dangerous situations.
In this context, analyzing the visual complexity of real-world displays with human evaluation in focus, can help understand how humans interpret the world around them. Using the gained knowledge of such analysis help stimulate human visual attention in AI-systems and in autonomous cars in particular, enabling more effective communication between humans and AI systems.
Previous studies[3] have addressed the ethical issues in designing AI systems, and aimed to use the gained insights from visual complexity analysis, visual clutter measures and quantifying visual features to construct and simulate driving scenarios that are particularly chosen to challenge the system with simulated, but similarly complex driving scenes that a human might face everyday.
Addressing these ethical issues is equally important to addressing the technical issues of human-computer interaction systems which is the reason behind this study and other related work that aim to provide standards and guidelines for designing human-centered AI systems.

1.1 Problem Formulation

The goal of this thesis is to conduct an analysis on visuospatial complexity of displays using existing computational models. The previously mentioned computational models, Feature Congestion and Subband Entropy, are implemented using Matlab and are introduced by researchers[4] from artificial intelligence, brain and cognitive sciences. The results of utilizing these computational models will be used to present an approach for measuring visual complexity of scenes based on human evaluation and image annotations.

The following sub-goals need to be completed in order to achieve the main goal of this thesis:

- Testing and evaluating the performance of two computational models for measuring visual clutter in displays and comparing their results. This includes preparing suitable data sets that illustrate real-world scenes mostly related to driving tasks.

- Preparing and annotating a data-set with adjusted object classes and objects attributes using an annotation tool. The annotations will be exported and used for introducing an approach into measuring visual complexity out of a human evaluation.
- Implementing required scripts to run tests on the models and present the results of their performance in human readable format.

1.2 Outline

The rest of this thesis is organised as follows:

- **chapter2**: An overview of previous works related to visuospatial complexity and visual clutter. An introduction to related terms and tools for visual complexity analysis and image annotations.
- **chapter3** : presents the used data set for utilizing visual complexity models and creating annotated labels. Additionally, describing the implemented scripts in Matlab and Python for data analysis and image processing.
- **chapter4**: Gives an outlooks on the results and presents the performed analysis together with the steps taken to utilize the computational models of visual clutter, as well as presenting the formula formed using annotated labels as an approach for measuring visual complexity of different driving scenes.
- **chapter 5**: A conclusion and summary of the study and future work.

Chapter 2
Related Works

2.1 Visuospatial complexity and visual features of displays

Visual complexity is described as the level of details or information that is present in an image[5] and is associated with the combination of visual and spatial characteristics that exist in visual displays[3]. The fact that the analysis of visual complexity refer to the study of visual features and explaining their affect on the visual attention, there is a need to present a model to quantify the visual features of displays in order to enable the process of measuring and comparing the complexity of images.

Kondyli et al. (2006) a Ph.D. student at the Center for Applied Autonomous Sensor Systems (AASS), suggested a taxonomy and model of visuospatial complexity model for visual features that exist in natural dynamic scenes.The model divides visual features that exists in dynamic scenes into three main categories, quantitative attributes, structural attributes, and dynamic attributes.
Quantitive attributes such as variety of colors, variety of objects, luminance and orientation refers to so called, low-level attributes that are considered as key features related to clutter of displays[4] which is presented and seen in Rosenholtz [1] measure models of visual clutter.

2.2 Visual clutter

Visual clutter refers to a state where an excessive amount of features or their representation[4] in the visual scene lead to crowding[6], negatively affects visual attention and decrease object recognition performance(Wolfe J.M. ,1996).
Several studies examined the effect of visual clutter on human visual perception by examining the influence of clutter on eye movements during scene perception[7], the contribution of clutter on visual attention such as searching for people in different visual scenes[8] or finding an object on a cluttered computer desktop (Wolfe J.M. ,1996) as well as the affect of clutter on reaction time during a driving-related task[9][10].
Visual clutter can be caused by the presence of numerous objects, complex shaped objects or a distracting background. Similarly, the representation of objects in a visual scene[1] have an impact on visual clutter. Grouping objects by size, color or by type is one method used to decrease clutter. A Research has shown that organization of this sort[11] has a positive effect on search performance in different visual tasks.

The three visual attributes: color, luminance and orientation are visual features that are related to clutter. Color is a feature that seem important for humans perception. Visual design guidelines for example, Microsoft visual design guidelines takes these observations on features leading to clutter as a guideline into the design of user-interfaces and warn

that many colors in a user interface causes clutter and distract the viewer from capturing important visual information.
Contrast energy have shown to be a factor that lead to high clutter, in addition to its role in the discrimination of salient vs non-salient images[12]. In a visual scene, objects with higher contrast energy, such as objects with high brightness or with a color difference from the background are more recognisable than objects with low contrast energy or similar color variation compared to their background.
The effect of orientation on visual clutter refers to the alignment of elements within a visual scene[4]. When elements in a scene have similar orientation or alignment such as lines, edges and objects, the scene appears more organized and less cluttered. On the other hand, when elements of a scene have different orientations the scene appears more visually complex due to clutter.

The following sections of this chapter represent two computational models introduced by Rosenholts (2007) for measuring visual clutter and how the models are developed to capture visual features contributing to visual clutter.

2.3 Measure of Visual clutter

2.3.1 Feature congestion measure of visual clutter

Feature congestion model is one of several models introduced by Ruth Rosenholts(2007), together with two other cognitive scientists for measuring visual clutter.
The models are developed to measure clutter in complex displays that simulate complex search tasks such as searching for an object between a set of similar objects on a table or in a room, or detecting a an object in images with different settings.
Feature congestion model measures three key features: color, luminance and orientation.By decomposing an input image into feature vectors(Rosenholtz, 2007) and then by combining the features at different scales the model produces three clutter maps representing the visual clutter of the input image "color congestion", "texture congestion" and "orientation congestion".

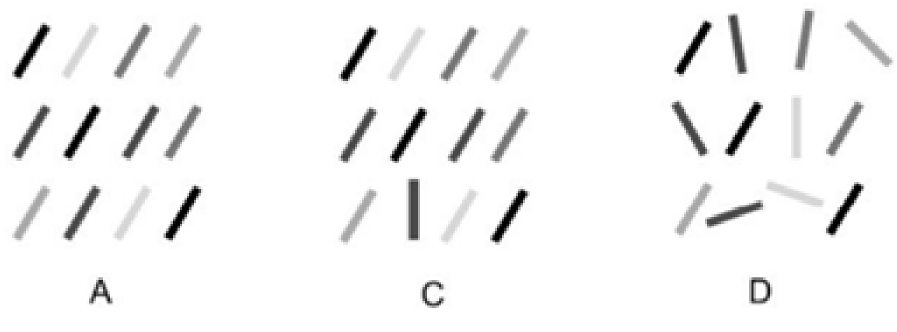


Figure1:the combination of luminance and orientation features.

The three different groups of items presented in figure1 illustrated in the study[1] indicates the affect of combining the two features, luminance and orientation on object detection.
A Group of items with high luminance and low orientation variability or low luminance and high orientation,as shown in sub-figures (A) and (D), makes it hard to detect an object using only these two features. On the other hand, an object with a different orientation than the objects withing the group,as visible in sub-figure(C) makes it easier to detect that object as it stands out from the group and draws the attention of the viewer.

Applying Feature Congestion model on natural displays from everyday situations captures the affect of these features and their affect on visual clutter in a similar way to the representation viewed in Figure1 .

2.3.2 Subband entropy measure of visual clutter

Subband Entropy clutter measure is based on the idea that visual clutter can be characterized by the amount of information contained in the sub bands of an image [1]. By decomposing an input image into sub bands of different spatial scales and orientations, the complexity is obtained by computing the entropy of each subband and further by combining the entropies of subbands to obtain a scalar value of visual clutter. The process of decomposing an image into subbands is inspired by the visual decomposition that occurs in human vision.
Subband entropy measure, in similar way to Feature congestion measure,implicitly captures the relation between clutter and high variability of features. A higher color variability contribute to more clutter, in similar notion, a higher entropy value for subbands of an image is translated to a higher value of clutter.

2.4 Image annotations as a tool for measuring visual complexity

Image annotations is the process of labelling raw data in form of images. The labels represent the objects that exist in an image or specific regions of interest in an image . By annotating images, the annotator provide metadata to a data set. The use of image annotations is common for computer science and specifically in the field of machine learning. Annotated data sets are used in supervised learning for AI algorithms that use these data sets for tasks such as image classification, image segmentation and object detection in a way that simulate the human way of detecting objects.

Annotation tools and technique have a wide variety and using the right annotation tools, the choice of a proper data set as well as the quality of annotations has a great effect on the project and the expected results. Even though annotation tools provide the required techniques to label and annotate images, the process involves several steps before and after the annotation task is performed. Some annotation tools requires pre-processing of images or extracting images from a video as a set of frames while other tools help the user to overcome few pre-steps by providing a structure for object classes that are meant to be used to label objects.

In this thesis, the annotation tool, Scalabel, is used to perfom image annotation on data sets. The steps required to create an annotation project with Scalabel are provided in the next chapter together with illustrations of the user-interface and available techniques in the annotation tool.

Chapter 3
Implementation

3.1 Introduction

This chapter provides an overview of the software implementations and preliminary steps undertaken to conduct an anlysis of visual complexity. Moreover, it outlines the process of image annotations and provides a detailed explanation of the annotation techniques employed to create a data set that captures human visual judgment and evaluation of visual complexity and visual clutter of various driving scenes.

3.1.1 Data set preparation

The selected data set for testing the measurement models consists of frames encoded from various driving task videos. The diverse range of scenes aims to assess the performance of the models on different scenarios with different complexity levels in terms of visual clutter.
The data set include a number of episodes representing a driving scene taken from a driver's perspective. The episodes depict night time driving tasks while others showcase daytime driving situations. Additionally, the number of objects appearing in the frames and the objects categories varies across episodes as shown in sub figures (a),(b),(c) and (d) . Certain episodes appear more cluttered than others with a number of cars, buses, motorcycles etc while others appear less cluttered with less objects appearing in the scene together with a variation in the episode's settings in term of luminance and emitted light from the scene.

(a) peisode1 (b) peisode2 (c) peisode3 (d) peisode4

Figure 3.1: example images from episodes of the data set

.

The variation in settings and in number of objects applies on the episodes despite whether it is a night or day driving task and it aims to get a variation in clutter measure while testing and evaluating the performance of the previously named clutter measure models. Figure 3.2 shows the number of episodes that the data set consists of together with the number of frames for each episode.

Data set "New Delhi"	
Episodes	Number of frames per episode
Episode 1	92
Episode 2	288
Episode 3	144
Episode 4	290
Episode 5	150

Figure 3.2: Data set episodes and frames.

3.1.2 Matlab scripts

Given that the measure models are implemented in Matlab and the function for both models takes a single image as an input, a custom Matlab code has been developed to loop through a set of frames of the format .png , .jpg or .jfif and run feature congestion and subband entropy functions on each frame in each episode. The output is stored in an Excel sheet, enabling convenient analysis through tables or charts.
Additionally, a script has been implemented to encode videos into individual frames. These frames are then organized as a list of images stored within a folder, ready to be utilized as input for the measure models.

3.1.3 Scalabel: open source web annotation tool

Scalabel enables image annotation and video tracking using different annotation techniques such as bounding box, polygon/polyline and cloud point bounding box. The tool provides a relatively simple user-interface and provide the user with documentations and other help material in form of demonstration videos and file example on how to use the tool and adjust the annotation space to fit the user requirements. When the tool is built on the user machine, related files and dependencies are provided in the tool environment. The files includes several examples illustrating the structure of files and type of input that are supported by Scalabel.

The tool is possible to build using a Docker image which makes it easy to run and set up the tool by creating a connection as a local host and pulling the tool image to user's Docker desktop.
The following figure illustrate the user-interface that first appears when establishing a connection on the user machine.

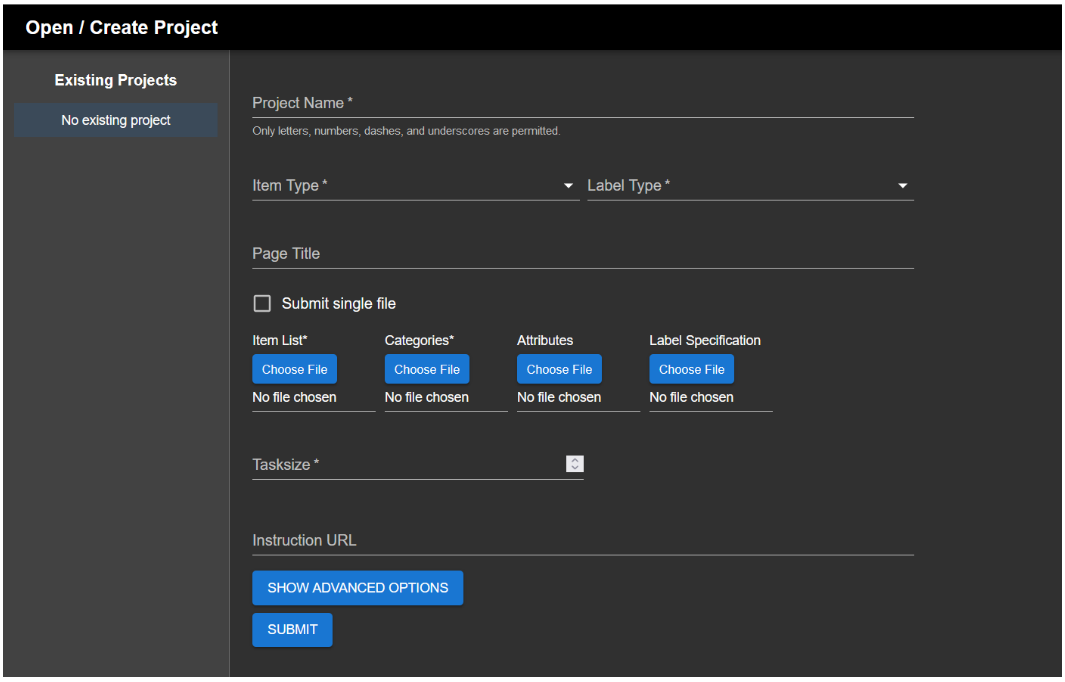


Figure3.3: Entry point of creating a project in Scalabel

As seen in Figure3.3, the entry point to Scalabel is reached by connecting to the local host localhost:8686

Created projects appear on the left side panel after filling in the required fields and uploading the required files. Each project most include a file that consist of an image list, a list of categories, list of attributes, and a list of labels if required. The files must be of specific format: .yml or .yaml files. Figures 3.4,3.5 and 3.6 below provides an example of the files containing the annotation project specifications.

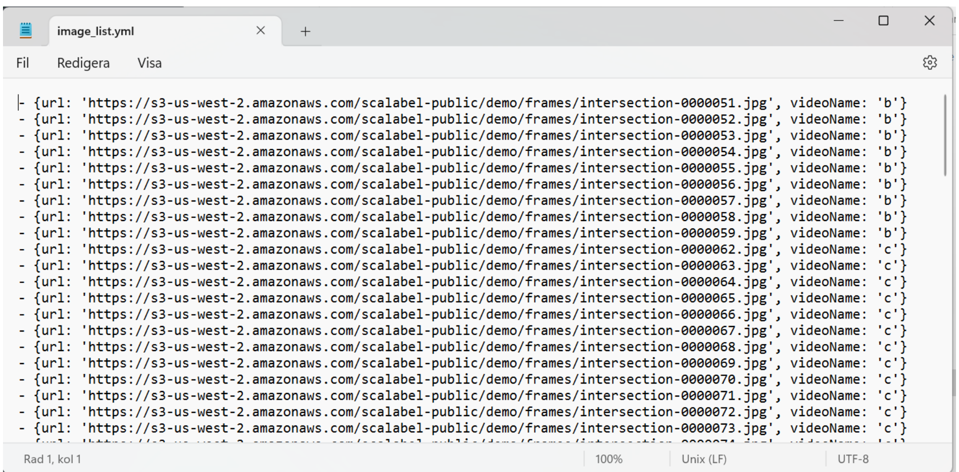


Figure3.4: Example of image list .yml file.

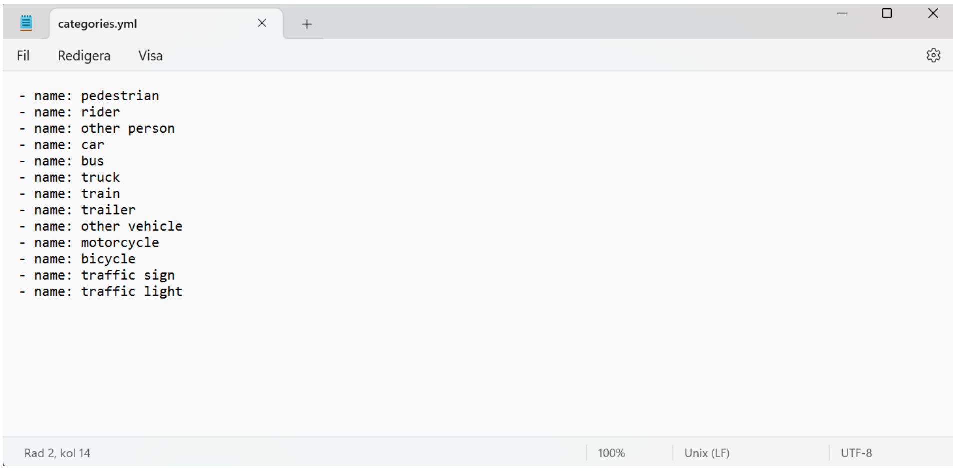


Figure3.5: Example of category list .yml file.

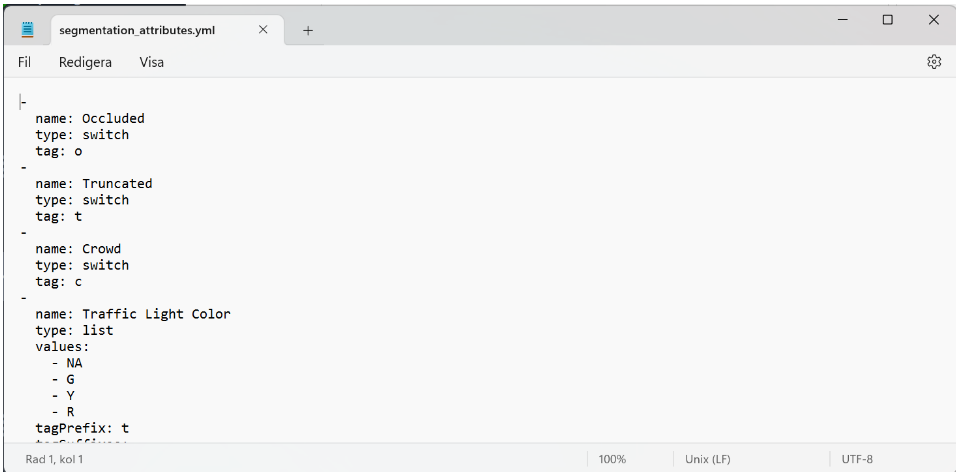


Figure 3.6: Example of attribute list .yml file.

Another alternative to uploading files for an annotation project in Scalabel is to upload a single JSON file with a list of frames/images, list of categories and a list of possible attributes and labels. This alternative can be considered as an efficient and less time consuming especially for larger data sets as it allows the user to create and use a single source file with all required specification for the annotated data set and is an easier solution for extracting values required for further analysis after annotating images.

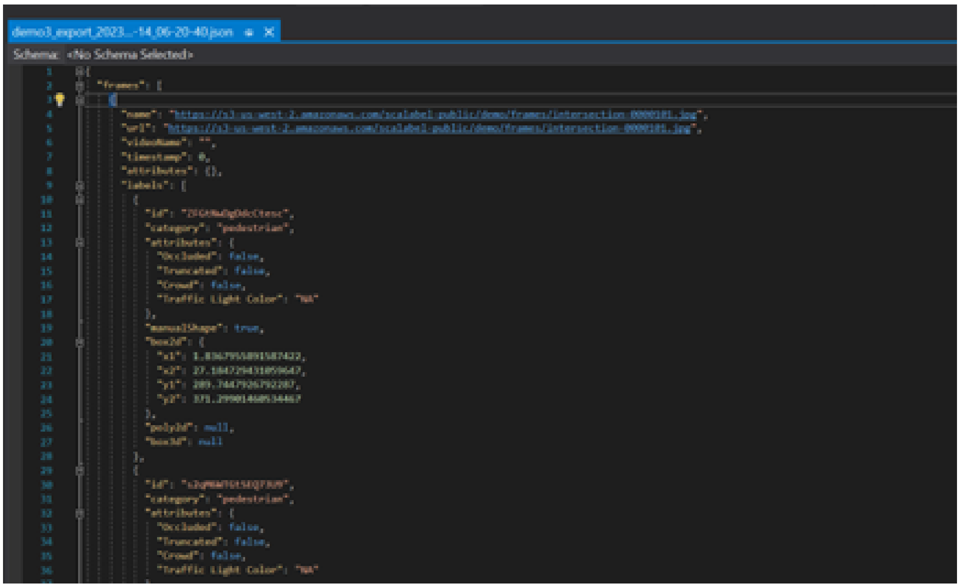


Figure3.7: Exported annotation file from Scalabel.

Figure3.7 illustrate the structure of an exported JSON file from Scalabel with a set of annotated objects. The file include all specification of the annotated data set together with the labels for each frame. For each frame in the annotated data set, there are a set of relative key values such as name, URL, and list of annotated labels. Labels associated with each frame are presented as a list with a number of key values such as the category of the label, label ID, a set of attributes and the coordination for the annotated label indicating the position of the label in the frame.

In the workspace, as illustrated in the Figure3.8, the left side panel displays the categories and attributes associated with the frame. The annotator selects the category and one or more attributes for the object to be annotated from this panel. Once an object, such as a car, is marked with a bounding box, the category name and a category ID are displayed on the left side of the bounding box.

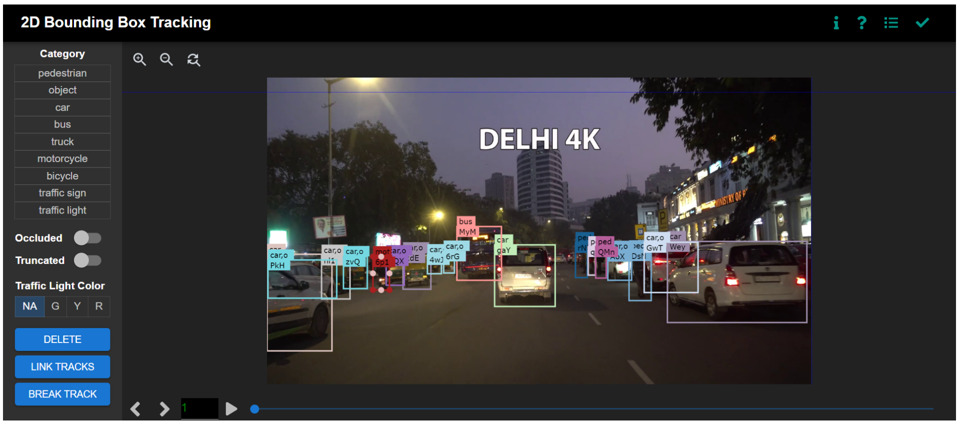


Figure3.8: Screenshot from Scalabel workspace.

Furthermore, there are two additional attributes available for objects: "Occluded" and "Truncated." These attributes provide extra information about the object. An occluded object refers to an object that is partially hidden by other objects, while a truncated object is not fully visible within the frame.By checking the corresponding checkboxes for "occluded" or "truncated" the attributes are indicated with a tag: "o" for occluded and "t" for truncated, next to the object's category name within the bounding box.

Scalabel supports object tracking and track linking. When an object is labeled in a key frame, the corresponding bounding box appears in subsequent frames with the same category and attributes. This feature allows the annotator to adjust the bounding box as the object moves or changes in size from one frame to another. In cases where an object disappears in a sequence of frames and reappears in a new frame, the bounding box can be linked to a previous one using the "track linking" option available in the left panel.

3.1.4 JSON files with image lists and labels

To simplify and speed up the process of image annotation, I have organized a JSON file for each episode included in the data set. These JSON files contain the specification required for annotating images, including categories, attributes, and labels for the image set or for that specific episode of the data set. Upon creating an annotation project, a single JSON file is uploaded into Scalabel as an alternative to separate files for image lists and their specifications.

3.1.5 Python scripts and libraries for data analysis

For the analysis of visual complexity of annotated images, a Python script has been implemented to extract key values from exported JSON files where Image labels are stored after the annotation process is completed. Using the data analysis library, Pandas, the script extracts the key values from the JSON files and stores output in an external file as a list of images together with the key values such as, category count, object count, occluded objects count etc.

Chapter 4
Results

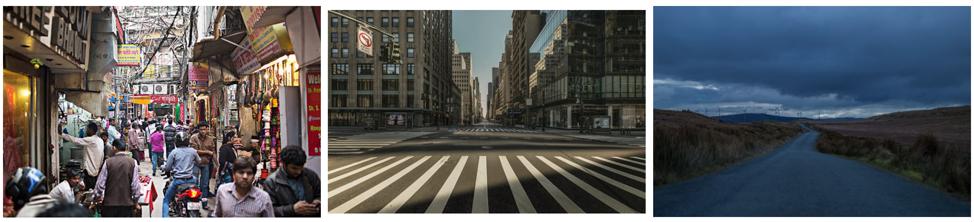
4.1 Visual clutter: Performance evaluation of clutter measure models

4.1.1 Pre-test of visual clutter measuring models

To prepare for the experiments, a preliminary test was conducted on a randomly selected images demonstrating a wide range of colors, variations in luminance and in number of objects as these features stimulate the models. These images were not limited to driving tasks, but involve diverse natural scenes.The aim was to obtain an understanding of how the model interprets and assigns values of visual clutter for the corresponding images and consequently, provide an insight into the expected values when applying the models on the driving data set.

Figure 4.1 depicts three showcases representing different levels of visual clutter, namely high, medium, and low. The values obtained from feature congestion model "clutterFC" and Subband entropy model "clutterSE" demonstrate how the combination of three key features: color, luminance, and orientation contributes to visual clutter.
In sub-figure (a), figure 4.1, the visual clutter is depicted through a combination of color variation and a significant number of objects present in the image. This combination results in a high value of "clutterFC," indicating a high level of clutter. Furthermore, the high value of "clutterSE" can be explained by the amount of information in the image, resulting a high entropy of image subbands.

When comparing the case presented in sub-figure (a) with the cases displayed in sub figures (b) and (c) of figure 4.1, it becomes evident that a decrease in color variety and the number of objects corresponds to a decrease in the value of visual clutter. This can be read in the values of running the measure models "clutterFC" and "clutterSE" on the images.



(a) clutterFC = 8.2999
clutterSE =3.7443

(b) clutterFC =3.1240
clutterSE =2.6294

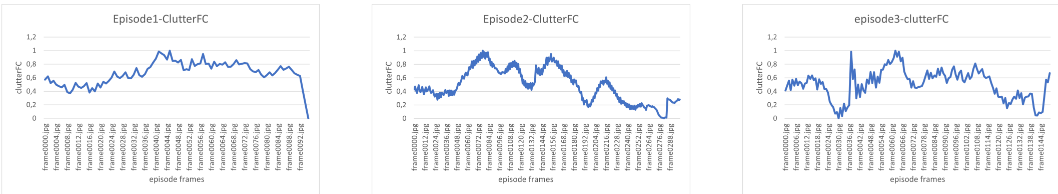
(c) clutterFC =1.4568
clutterSE =2.0485

Figure 4.1: High, medium, and low clutter displays

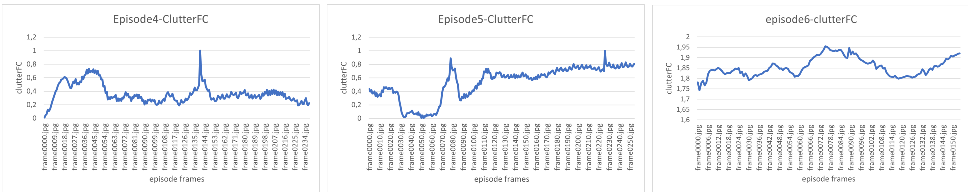
In sub figure (c), it is particularly noticeable that both models indicate a very low clutter value. This can be explained in the low contrast-energy and low orientation observed in the images.
It is important to note that using a single image does not serve as a sufficient measure of visual clutter of driving scenes. In driving scenes,the frame's settings and the feature combinations tend to change from frame to frame. Thereby, in the following sections of this chapter,the presented experiments involves applying the measure models on a data set consisting of a set of episodes with a number of frames. The output of the measure models is then presented for a whole episode.

4.1.2 Experiment 1: Feature congestion clutter measure

In this experiment, each episode of the data set is given as an input to the measure model in form of an image list. After running the measure function on the image list, the output results, represented as decimal values, are stored together with the corresponding frame in an Excel sheet. The results are represented using 2D-line charts. The values of clutter represented on the x-axis and the corresponding frame on the y-axis. The end of this section represent a table with "clutterFC" score for each episode based on the performance and values of Feature Congestion model.



(a) episode 1 (b) episode 2 (c) episode 3



(d) episode 4 (e) episode 5 (f) episode 6

Figure 4.2: Feature congestion performance on data set episodes

As seen in Figure 4.2, the output of the measure model and the value of clutter for each episode is represented by a two dimensional line chart. The x-axis (clutterFC) shows how the values and intervals varies across episodes. Additionally, it is noticeable that the level of visual clutter does not only changes between episodes, but also within frames of a specific episode.
For instance, in sub-figure (c) illustrating clutter measure in episode 3, the values fall within the range of 1.77 to 1.88. On the other hand, in sub-figure(d) illustrating clutter measure in episode 4, it is noticeable that the range of clutter values is greater, ranging from 1.9 to 2.11 .This suggests that Episode 4 is characterized by a higher level of clutter and episode 3 being the episode with lowest visual clutter when compared to the other six episodes.

The performance of the measure model and the results associated with each episode can be explained by taking a closer look into the driving scene that each episode depicts. Considering episode 2, which has the highest range of visual clutter across all six episodes,

the episode depict a daytime driving scene with a significant amount of objects that are not similar in shape or color. In comparison with episode 3 that is characterized by lower amount of objects and darker theme which explains the range of values seen in the corresponding 2D line charts for each episode in sub-figures 4.2.b and 4.2.c .
The visual features are even changing throughout the duration of the episodes. Figure 4.3 and figure 4.4 illustrate the frames with highest and lowest clutter value in episode2 and episode 3.The left frame illustrate the highest cluttered frame in the episode and the right figure illustrating the least cluttered frame according to their clutter value measured by Feature Congestion model.

After analyzing the performance of Feature Congestion model on the data set, and examining the results presented in Figure 4.2, a complexity score for each episode is given by assigning a number between 1 and 6 for each episode. Score 6 being the highest score is assigned to the episode with most clutter, determined by the range of values obtained from running the model on that specific episode. similarly, a score of 1 is assigned to the episode with the least clutter (lowest range of values).

Figure 4.3: Highest cluttered (left side frame) and lowest clutter (right side frame) frames in episode2 according to FC model

Figure 4.4: Highest clutterd(left side frame) and lowest cluttered(right side frame) frames in episode3 according to FC model.

The following table illustrate the score assigned for each episode according to the performance and results of Feature Congestion model.

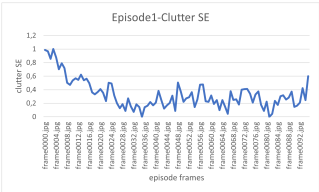
Episode	Clutter_FC Score
1	5
2	6
3	1
4	3
5	4
6	2

Clutter score based on Feature congestion measure of clutter

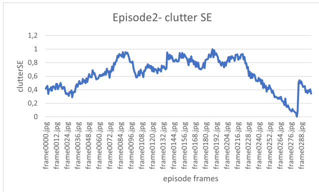
4.1.3 Experiment 2: Subband entropy clutter measure

The same method for running Feature congestion model on each episode has been used to perform experiment2 for running the second measure of visual clutter, Subband entropy(SE) measure. Figure 4.5 illustrates the results of running SE on the same data set.

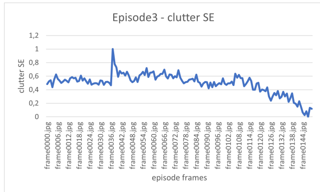
From figure 4.5 one can draw conclusion that episodes which has been assigned a high score in experiment 1 with Feature congestion model, are also characterized by high clutter ranges and values according to Subband Entropy model. Episode 2 is still the episode with highest clutter between all episodes and episode three is still among the episodes with lowest ranges of clutter values according to the second measure, Subband Entropy measure.



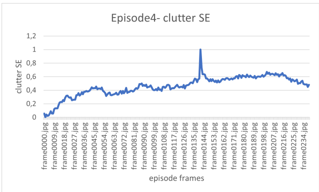
(a) episode 1



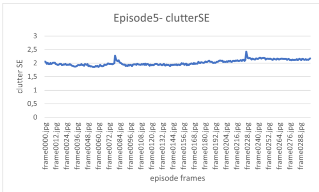
(b) episode 2



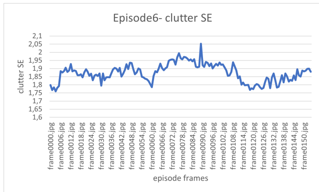
(c) episode 3



(d) episode 4



(e) episode 5



(f) episode 6

Figure 4.5: Subband entropy performance on data set episodes.

The following table illustrate the scoring of episodes based on the performance of Subband Entropy measure, with score 6 assigned for episode2 with highest complexity and score 1 assigned for episode 6.

Episode	Clutter_SE Score
1	3
2	6
3	2
4	5
5	4
6	1

clutter score based on Subband entropy measure of clutter

It is noticeable that the score assigned for each episode does not change dramatically between experiments1 & 2 that utilize visual clutter measures as both measures capture similar aspects of visual clutter, which can be defined as the amount of visual information contained within a scene.

Across both experiments, episode 2 consistently maintains the highest score, indicating the highest level of clutter among all episodes while episodes 3 and 6 consistently receive the lowest scores, signifying lower clutter levels.However, there are differences in the scores assigned to episode 1 and episode 4 when comparing the Subband Entropy measure to the Feature Congestion measure where episode 1 has been assigned a lower score and episode 4 a higher score compared to their previous score based on Feature Congestion measure.
The difference can be explained by examining the frames of episode 1 and episode 4. In episode 1, it is clear that the frames include a substantial amount of visual information and high variation in the visual features, but the scene and its features does not remarkably change between frames as its visible in figure4.6 .
The least cluttered frame on the left side and most cluttered frame on the right side ap-

pear to be similar. On the other hand, in episode 4, the amount of energy or emitted light changes remarkably as the scene goes on and the location and type of objects changes as well. Taking a close look into figure4.7, one can see that the left frame which has been given lowest clutter value by Subband entropy measure seems to be darker and less cluttered than the frame on the right side with a significant change in the scene where the frame appears more cluttered and the amount of emitted light is higher compared to the frame on left side with low clutter value.

Overall, while there may be slight variations in the scores assigned by different clutter measures, the underlying characteristics of the scenes remains consistent.

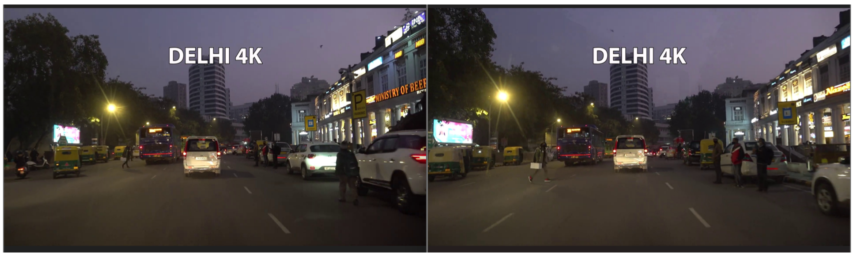


Figure 4.6: Lowest(left frame) and highest(right frame) cluttered frames in episode 1 according to SE model

Figure 4.7: Lowest(left frame) and highest(right frame) cluttered frames in episode4 according to SE model.

4.1.4 Experiment 3: Visual complexity measure of annotated images

The previous experiments provides valuable insights into the measurement of visual complexity of displays using different techniques, but based on similar analysis of visual complexity that takes into consideration the combination of visual features and their representation in the image. These findings can be used to explore alternative approaches for measuring visual complexity, employing similar scientific principles and formulas.

Building upon this foundation, Experiment 3 aims to introduce an approach for measuring visual complexity of annotated images .Using annotation techniques to label objects that are relevant to the driving task and building upon human judgement for evaluating the importance of each object to the measure of visual complexity a simple formula is used to measure visual complexity of each episode in the annotated data set.

To conduct this experiment and create an image annotation project, the same data set used in Experiments 1 and 2 will be employed. Annotation categories and attributes are chosen to represent objects of the frame. The objects are labeled using a 2D bounding box. This technique involves marking bounding boxes around all objects that are visible within the car driver's field of vision or are objects that are relevant to the driving task such as pedestrians on the road, cars on road, moving motorcycle and other vehicles etc.

Figure 4.8: Example annotated frame from the data set.

Figure 4.8 shows an example from the annotated data set, specifically episode 2. The colored boxes around the objects such as "car", "motorcycle" are marked as a 2D bounding box first by choosing the category of the object and then by checking the attribute box for "Occluded" if the objects is partly or fully hidden behind another object and "Truncated" in case an object that is being labeled does not completely appear in the frame. The number of annotated objects and their attributes are used in the formula presented below for measuring visual complexity of annotated images.

The suggested formula for calculating visual complexity of annotated episodes and frames:

CS= log2(sum(cat_n*cat_w)/NI * (NI-NO)/Nf .

CS: complexity score.
Cat_n: number of annotated categories.
Cat_w : weight of category.
NI: number of annotated instances (occluded + non occluded).
NO: Number of occluded instances.
Nf: number of frames per episode which complexity.
score is to be measured.

The formula is applied on frame levels, meaning that a complexity score is calculated for each frame in an episode, and later a complexity score is calculated for the whole episode. The only modification that applies on the formula when calculating complexity score for an individual frame is diving the term (NI-NO) by 1 instead of NF "number of episode frames". Basically eliminating the step of dividing by the number of frames as the score is calculated for a single frame.

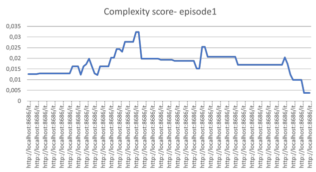
The categories weights "cat_w" are chosen based on a human judgement giving each category a weight based on its importance in the context of a driving task scene. For example, an annotator and out of driver perspective, might consider a driving scene with a significant amount of pedestrians to be more complex than a scene with a number of cars and other vehicles as the existence of unguarded humans between cars is considered more of a critical safety situation. Given that information, the category "Pedestrian" would get the highest weight between all categories used in the annotated frames.

The following table illustrate the assigned weights for each category based on an individual judgement and out of safety point of view:

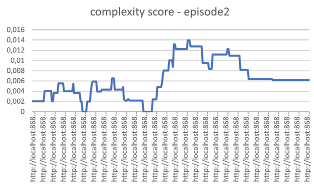
Category	Category weight
Pedestrian	5
motorcycle	4
Car	3
Bus	2
Truck	1

categories of driving scene with their assigned weights.

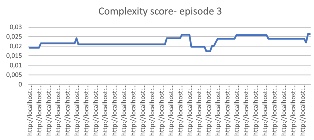
Gifure 4.9 illustrate the results of calculating complexity of each episode using the complecity score formula in a 2D line charts.
On the x-axis one can read the range of complexity score for each frame and then by plotting a 2D line chart, represent the complexity score of a whole episode.



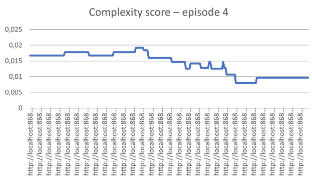
(a) episode 1



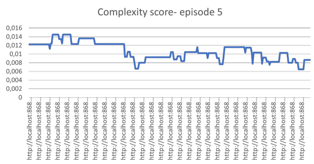
(b) episode 2



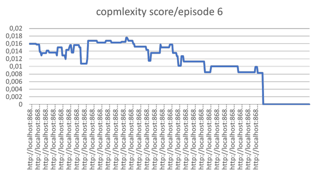
(c) episode 3



(d) episode 4



(e) episode 5



(f) episode 6

Figure 4.9: Complexity score for each episode in the annotated data set.

A summary for the results of applying complexity score formula on the annotated data set together with the specification of each episode that has been annotated and used for measuring visual complexity.

Annotated episodes/images	categories	NI	NO	CS
Episode 1: 102 frame	4: pedestrian, motorcycle, car, bus	5-13 per frame	3-9 per frame	6
Episode2: 299 frame	2: motorcycle, car	4-9 per frame	1-7 per frame	1
Episode 3: 199 frame	4: pedestrian, motorcycle, car , bus ,	4-10 per frame	2-3 frames	5
Episode 4: 149 frame	2: pedestrian, car	3-6 per frame	0-2 per frame	4
Episode5: 299 frames	4: Pedestrian, motorcycle,car, truck	6-10 per frame	1-4 per frame	2
Episode 6: 299 frame	5: pedestrian, motorcycle, car, bus , truck	5-10 per frame	0-7 per frame	3

Visual complexity score of annotated episodes.

Episode 1, has 5-13 labeled instances from 4 present categories "pedestrian","motorcycle", "car" and "bus". The number of occluded objects per frame is on average between 3-9 occluded objects per frame which explain giving episode 1 the highest score among the six episodes. The episode with the lowest score, episode 2 have no labeled objects of category "pedestrian" with the highest weight among all categories and with only 7-9 labeled instances of two categories "motorcycle" and "car". The number of occluded objects is clearly lower with a single occluded objects per frame and up to 7 occluded objects.

Chapter 5
Conclusions

The results of visual complexity analysis using different methods presented in Chapter 4 offers a significant knowledge regarding the impact of visual clutter and the visual information contained in an image on the level of clutter on the visual complexity of an image. The visual clutter models demonstrate consistent results when applied to the same data set, providing strong evidence of how combinations of features such as color, luminance, and orientation contribute to higher levels of visual clutter. These results, combined with existing scientific researches on visual complexity, offers a foundation for standardizing the development of multi modal interaction AI systems and can be used as methods for improving the system's performance to achieve effective and safe communication between humans and machines.

Scalable and other annotation tools offers opportunities for creating reliable measures of visual complexity by utilizing the output of annotated images. These tools enable the annotators to adjust and control the specification of an annotated image to highlight important and related information contained in the annotated images. For example, one approach discussed in this thesis involves measuring visual complexity based on the number of categories, number of labeled instances and the ratio of occluded objects to the total number of labeled instances. However, other approaches are possible to formulate considering other aspects of annotated images, such as the size of labeled instances, their visibility and their distance from the fixation point. These information can be used to implement alternative measures of visual complexity for the purpose of creating training data sets for machine learning, multi modal computer interaction systems or other applications concerned with the visual complexity of displays.

References

[1] Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. Feature congestion and subband entropy measures of visual clutter. 2007. (Cited on pages 1, 5, 6, and 7.)

[2] Scalabel: A versatile web-based visual data annotation tool. (Cited on page 1.)

[3] Jakob Suchan, Mehul Bhatt, Srikrishna Vardarajan, Seyed Ali Amirshahi, and Stella Yu. Semantic analysis of (reflectional) visual symmetry: A human-centred computational model for declarative explainability. *Advances in Cognitive Systems*, Vol 6:65–84, 2018. (Cited on pages 3 and 5.)

[4] Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. Measuring visual clutter. *Journal of vision*, 7(2):17–17, 2007. (Cited on pages 3, 5, and 6.)

[5] Joan G Snodgrass and Mary Vanderwart. A standardized set of 260 pictures: norms for name agreement, image agreement, familiarity, and visual complexity. *Journal of experimental psychology: Human learning and memory*, 6(2):174, 1980. (Cited on page 5.)

[6] James A. Stuart and Hermann M. Burian. A study of separation difficulty*: Its relationship to visual acuity in normal and amblyopic eyes. *American Journal of Ophthalmology*, 53(3):471–477, 1962. (Cited on page 5.)

[7] Tom Foulsham, Joey T. Cheng, Jessica L. Tracy, Joseph Henrich, and Alan Kingstone. Gaze allocation in a dynamic situation: Effects of social status and speaking. *Cognition*, 117(3):319–331, 2010. (Cited on page 5.)

[8] Krista A. Ehinger, Barbara Hidalgo-Sotelo, Antonio Torralba, and Aude Oliva. Modelling search for people in 900 scenes: A combined source model of eye guidance. *Visual Cognition*, 17(6-7):945–978, 2009. PMID: 20011676. (Cited on page 5.)

[9] William Horrey and Christopher Wickens. Driving and side task performance: The effects of display clutter, separation, and modality. *Human factors*, 46:611–24, 02 2004. (Cited on page 5.)

[10] Edquist, jessica (2019). the effects of visual clutter on driving performance. monash university. thesis. (Cited on page 5.)

[11] Anne Treisman, Daniel Kahneman, and Jacquelyn Burkell. Perceptual objects and the cost of filtering. *Perception psychophysics*, 33:527–32, 07 1983. (Cited on page 5.)

[12] Dashan Gao and Nuno Vasconcelos. Decision-theoretic saliency: Computational principles, biological plausibility, and implications for neurophysiology and psychophysics. *Neural computation*, 21:239–71, 01 2009. (Cited on page 6.)