UNDERSTANDING PERCEIVED SENSE OF MOVEMENT IN STATIC VISUALS USING DEEP LEARNING

by

SHRAVAN KALE

A THESIS

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

September 2018

THESIS APPROVAL PAGE

Student: Shravan Kale

Title: Understanding Perceived Sense of Movement in Static Visuals Using Deep Learning

This thesis has been accepted and approved in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science by:

Dejing Dou	Chair
and	
Janet Woodruff-Borden	Vice Provost and Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded September 2018

THESIS ABSTRACT

Shravan Kale

Master of Science

Department of Computer and Information Science

September 2018

Title: Understanding Perceived Sense of Movement in Static Visuals Using Deep Learning

This thesis introduces the problem of learning the representation and the classification of the perceived sense of movement, defined as dynamism in static visuals. To solve the said problem, we study the definition, degree, and real-world implications of dynamism within the field of consumer psychology. We employ Deep Convolutional Neural Networks (DCNN) as a method to learn and predict dynamism in images. The novelty of the task, lead us to collect a dataset which we synthetically augmented for spatial invariance, using image processing techniques. We study the methods of transfer learning to transfer knowledge from another domain, as the size of our dataset was deemed to be inadequate. Our dataset is trained across different network architectures, and transfer learning techniques to find an optimal method for the task at hand. To show a real-world application of our work, we observe the correlation between the two visual stimuli, dynamism and emotions.

CURRICULUM VITAE

NAME OF AUTHOR: Shravan Kale

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR
Vidyalankar Institute of Technology, University of Mumbai, Mumbai, India

DEGREES AWARDED:

Master of Science, Computer and Information Science, University of Oregon, 2018
Bachelor of Engineering, Computer Engineering, Vidyalankar Institute of Technology, 2015

AREAS OF SPECIAL INTEREST:

Machine Learning, Deep Learning, Computer Vision

PROFESSIONAL EXPERIENCE:

Web Developer and Co-Founder, Beyond The Byte, 2012-2015

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support of my family, advisors, and friends. I would like to thank my advisor, Professor Dejing Dou for his guidance and encouragement, Professor Aparna Sundar and Professor Conor Henderson for their ideas, discussion, and insights into the field of consumer psychology. I would also like to thank Nisansa deSilva for his vital suggestions. Lastly, my family and friends, for standing by me, every day.

For my parents and friends, a large family across two continents

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

We introduce the problem of learning the representation and classification of images that have a perceived sense of movement. The movement is defined as dynamism and its absence is defined as still. Dynamism is a stimulus in static visuals as it affects the retention of attention towards the said visual. It induces a sense of motion which engages the viewer to a visual in which the motion is only implied. The study of this engagement, its improvement, and the stimuli that affect it are of interest in the field of marketing and psychology. We intend to study it to emulate the human understanding of the presence or absence of dynamism using Deep Learning. Our image classification task is different from the popular Object Recognition (OR) task since OR deals with recognizing the category of the object using its physical properties. Our task consists of understanding and recognizing the perception of the sense of movement and its relative absence. Affective Image Classification (AFIC) is another domain that we consider similar to the task of our domain since it includes the study of emotion as another stimulus affecting the viewer of a static visual. From AFIC we understand that along with the category of objects in the movement, the mood, personality and even the environment of the viewer affects the emotion evoked by an image. Though dynamism is studied only with respect to the movement. The rest of the probable properties of images and the perception of the images by the viewer are left to human intelligence, verified by the effect of dynamism. Towards the further understanding of dynamism, we study the novel task of classification of the presence and absence of movement in images using Deep Learning.

Problem Definition

Within the literature of the field of consumer psychology, Dynamism is seen as a tool to increase consumer engagement with static visual (or images) used for marketing products or services by various brands. The emergence of marketable platforms such as social media and brand-specific websites has led to the monetizing of consumer engagement which values the retention in engagement. Since dynamism is relative in nature, researchers and the industry have always turned to survey groups to determine its presence or absence and its degree. A considerable effort as seen in the study by Cian et al. (2014) is required to determine the dynamism in existing images and the new images that an artist or marketers create to increase the said engagement. There is a requirement to make this process more efficient, accessible and robust along with defining a method of understanding said classification, the factors that dynamism is affected by and the factors it directly or indirectly affects.

To meet some of these requirements we turn to Artificial Intelligence, specifically, image classification using Deep Learning. The goal is to construct (train) a classifier that has learned the ability to detect dynamism, and its degree. Such a model can be made easily available to artists or marketers that may or may not have resources like a survey group of a needed kind, size and/or even the complete domain knowledge of dynamism. These models can be made efficient and robust as they can be created for a larger and general target audience or even a smaller specific target audience. The idea is to create a generic model initially that can be then trained on an image dataset with properties as required by the task. Given the construction of such a model, it would enable us to study the effect of dynamism on other properties of images such as the emotions evoked by an image.

Looking further ahead, such models could spur studies in understanding the factors that affect dynamism, the methods of understanding the reasoning behind the classification and even the generation or addition of dynamism and its degrees to images that lack it.

Solution Approach

The first approach considered towards this problem was the classical machine learning approach but it would have required different handcrafted features required for the said approach. The understanding of such features would require an additional study in the features from other domains such as psychology. Even if such set of features were obtained, they would be lower level features compared to what a DCNN would obtain. Hence we chose Deep Learning, specifically DCNN, due to its excellent ability to extract higher-level abstract features which we intend to exploit since dynamism cannot be distinguished based only on the lower level features such as edges and shapes.

Due to the novelty of the study such datasets are not available and even if constructing such a dataset is attempted, it is not yet feasible to construct it in the magnitude of ImageNet Deng et al. (2009a) which is a requirement for the efficient training of millions of parameters of a DCNN.

To tackle the above-mentioned problem, we look at the concepts of Data Augmentation in the study by Krizhevsky et al. (2012). Augmentation helps for synthetically increasing the size of the dataset, while maintaining type variation and adding spatial invariance. These methods assist to improve the performance by adding more data. Then to further compensate for the lack of a large dataset and over-fitting, we look into the concepts of Transfer Learning as a method to

transfer knowledge(or lower level features) from a task in another domain to the one mentioned in our problem.

Outline

This thesis will delve into the definitions and important concepts of the above-mentioned problem and solution. Followed by, explanations for the list of experiments, their results, and observations. We would conclude with suggested scalable improvements and possible future work.

CHAPTER II

PERCEIVED SENSE OF MOVEMENT

Concept of Dynamism in Consumer Psychology

The definition of the term dynamism in the context of consumer psychology is the perceived sense of movement in static visual or images as mentioned in Cian et al. (2014). This definition leads to a term still for the absence of a perceived sense of movement. This concept is studied in consumer psychology because consumers are the recipients of dynamism as a stimulus when they view visuals such as images which are meant to be engaging in the manner of advertisements or posters. The effect of this stimulus is measured with respect to the engagement and attitude they have towards the target object or associated brand in the said visual. In our quest to build a DCNN which has the ability to distinguish between images with or without the perceived sense of movement, we found the study by Cian et al. (2014) on the effect of dynamism on static visuals, specifically brand logos, in terms of consumer engagement and the consumer attitude towards the brand. We study the various experiments in Cian et al. (2014) to understand the types of dynamism and its effects.

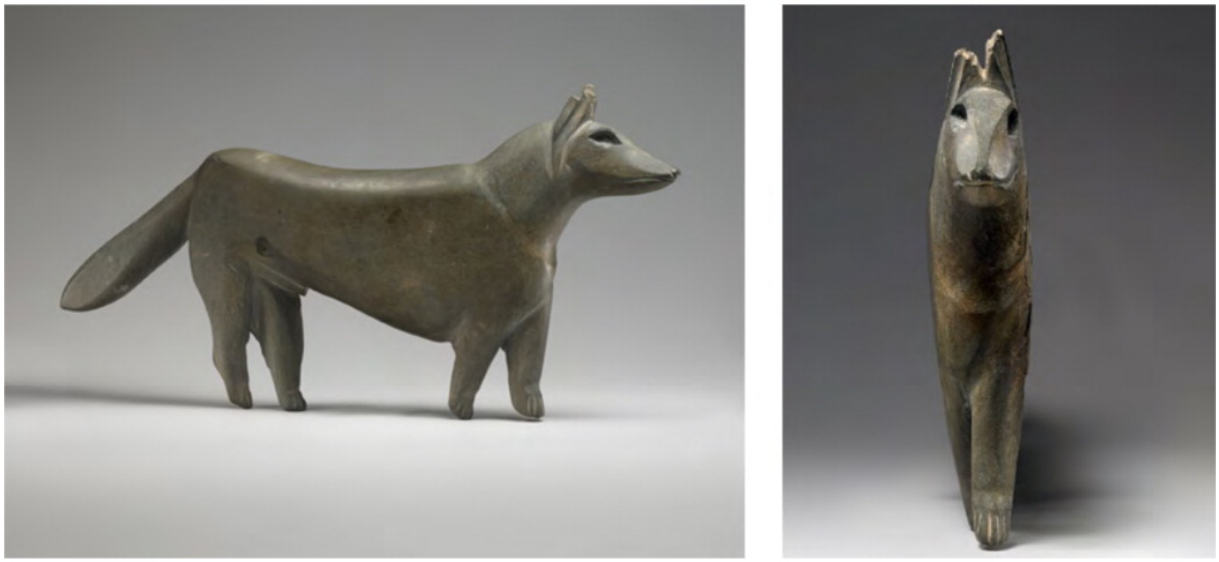

Image source: Phoebe A. Hearst Museum of Anthropology, University of California, Berkeley.

FIGURE 1. Jackal Sculptures from Cian et al. (2014)

Dynamism as a stimulus has been around since the humankind learned to depict art in the form of paintings and sculptures as mentioned in Cian et al. (2014). An excellent example as seen in figure 1 is that of a sculpture of a jackal that seems to have been frozen in motion. Even though the sculpture is an inanimate object, the sculpting of the foxs lifted tail and separated legs lead to the perception of motion.

FIGURE 2. Dynamism examples from Pavan et al. (2011)

The pictures in figure 2 is another example of dynamism. The image of the leopard in A is considered dynamic as it captures the animal in motion whereas the image of the dog is still due to its stationary position. The image of the goalkeeper in B is relatively still compared to the baseball player in A.

Previous Literature

Most of the literature that was published before Cian et al. (2014) focussed on only the comparison of the concept of dynamism, its absence, and the precursor to the perception of movement. An interesting definition is given in the study by Cian et al. (2014) defines Still images as the ability of the brain to generate representations of stationery and fixed objects that facilitate the recognition of the figure in the said image and the judgement about the objects visual properties whereas dynamic images as a representation of objects in implied motion such that the brain simulates the motion. This is an appropriate representation of

our experiment since our DCNN would create representations of these images and distinguish between them similar to the human mind. The literature such as Leborg (2006) and Dondis (1974) from the fields of art and design includes dynamism along with features of objects such as shape, color, and texture are known as the visual grammar. In section 3.1 we discuss the adaptation of these features as a classical machine learning experiment in a different domain.

Hypothesis

One of the hypothesis suggested in Cian et al. (2014) is that dynamism is directly proportional to the engagement with an image. It is so because the viewers of said image are able to imagine the implied motion in dynamism such that it holds their attention longer than a comparative still image. The rationalization behind the hypothesis is that the bounds of human imagination supersede a stimulus provided by the creator of that image. The authors also state that engagement is proportional to the attitude towards the brand related to the said image as proved by Pieters and Wedel (2007) and Teixeira et al. (2012). They add to the statement hypothesizing that dynamism in images is also proportional to the attitude towards the brand due to their proportionality with engagement. Although, they also mention some exceptions to the above-mentioned proportionality such that if dynamism is inconsistent with the characteristic (eg. modern or traditional oriented) of the brand termed as congruency, then the proportionality may not be maintained.

Dynamism, Its Types, and Effects

Various studies are conducted by Cian et al. (2014) to prove the above hypothesis. Since these studies do not use Deep Learning or any Artificial Intelligence algorithm, they rely on human survey groups for analysis. They conduct a pre-test for selecting images such that they have distinct higher and lower dynamism by keeping other variable characteristics constant for the experiment.

Studies on Dynamism

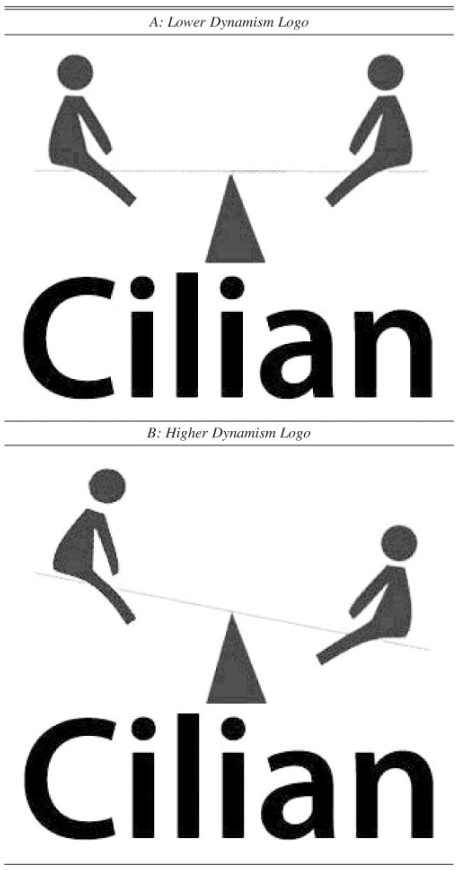

FIGURE 3. Relative dynamism from Cian et al. (2014)

For the pretest, two logos of a fictitious brand are created instead of a real brand to make sure there is no bias against a known brand which might affect the outcome of the experiments. The logo with the seesaw at equilibrium has a lower dynamism and the logo with the seesaw at an angle frozen in motion has higher dynamism as seen in figure 3. The first survey was conducted to make sure that the other factors such as visual appearance, complexity, informativeness, familiarity, and novelty had the least difference on a custom rating scale. The second survey was done on a two-item scale, the amount of movement seen in the logo and its dynamism. The pre-test images were concluded to have a significant difference in dynamism.

Evoked Dynamism and Attitudes

A study in Cian et al. (2014) was conducted to prove the hypothesis that higher dynamism leads to a better attitude towards the brand in the test. The attitude towards a brand was also rated on a custom scale. The test concluded that the survey group reports a better or more favorable attitude towards the brand with a higher level of dynamism. As a manipulation check, the survey group separate from the one in pre-test was asked to rate the logos on dynamism similar to the pre-test. The test showed similar results in all the studies mentioned further.

Types of Dynamism and Mediation through Engagement

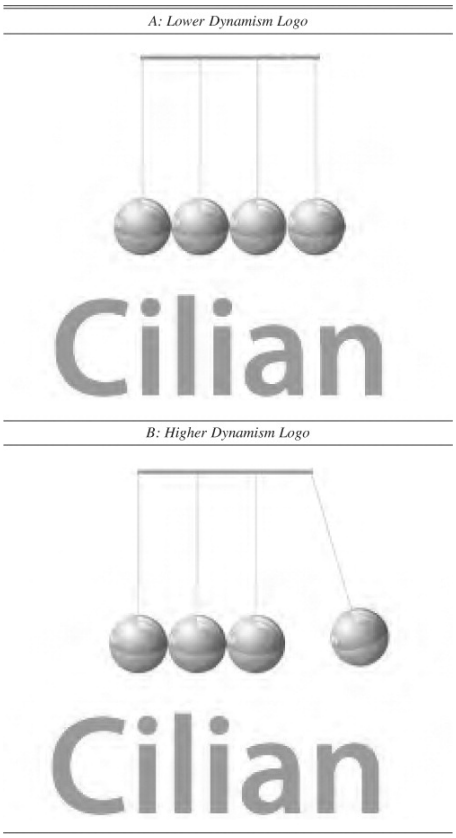

FIGURE 4. Types of dynamism from Cian et al. (2014)

The effect of dynamism on engagement and its effect on attitude towards the brand is gauged in Cian et al. (2014). Two pairs of logos with a different type of dynamism ie. frozen motion as seen in figure 4 and friction as seen in figure 5 is used. For frozen motion, a logo consisting of a Newton's cradle is used wherein the one with lower dynamism is stationary and the with higher dynamism is frozen in motion. Pretest similar to the section 2.2 was conducted for this study as well. The new logos were put through the same test as in section 2.2 and an additional test was conducted for engagement. The scale for engagement from Craig Lefebvre et al. (2010) was used comprising of involvement, engagement, how boring and

stimulating each image measured on a custom scale. The result of the study proved the hypothesis that dynamism leads to higher engagement and a favorable or better attitude towards the brand.

The hypothesis of engagement acting as a mediator for dynamism and attitude towards the brand mentioned in Cian et al. (2014) is proven by using mediation analysis. With controlled dynamism, engagement had a proportional effect on attitude towards the brand, but with controlled engagement, dynamism did not have a significant impact on the attitude towards the brand directly.

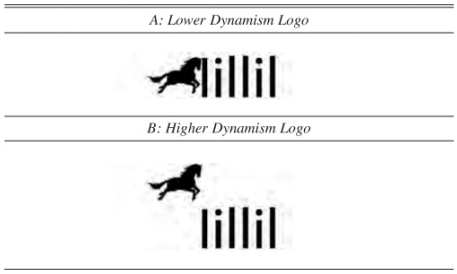

FIGURE 5. Dynamism as friction from Cian et al. (2014)

For friction, a pair of logos in figure 5 consisting of a horse and a text is used wherein the one with lower dynamism has the horse and the text in contact (friction) and the one for the higher dynamism has the horse and text separated by some arbitrary distance. Pretest similar to pretest in section 2.2 was conducted for this study as well. The survey group tested for attitude, engagement, and dynamism similar to the previous test. The test concluded with the result that dynamism was proportional to engagement and to the attitude towards the brand. Akin to how in the previous test, engagement played a mediating role in dynamism and attitude. The above tests conclude that the different types of dynamism does not affect the role of engagement as a mediator and proves the above-mentioned hypothesis.

Effect: Moderation by Congruence

A study by Cian et al. (2014) was designed to prove that the congruence between dynamism and characteristics of the brand moderates the effect of dynamism on attitude towards the brand. The Newtons cradle logos were used again wherein the company with a less dynamic logo was given a traditional and classical music orchestra description whereas for the company with a more dynamic logo was given a modern music orchestra description. After conducting pretests similar to section 2.2 the survey group evaluated for attitude towards the brand, engagement, and dynamism. The analysis on the evaluation proved that higher dynamism along with a modern description and lower dynamism with traditional description showed a higher level of engagement hence proving the moderation that congruence offers on the effect of dynamism and the attitude towards the brand.

Effect: Direction of Movement And Congruence

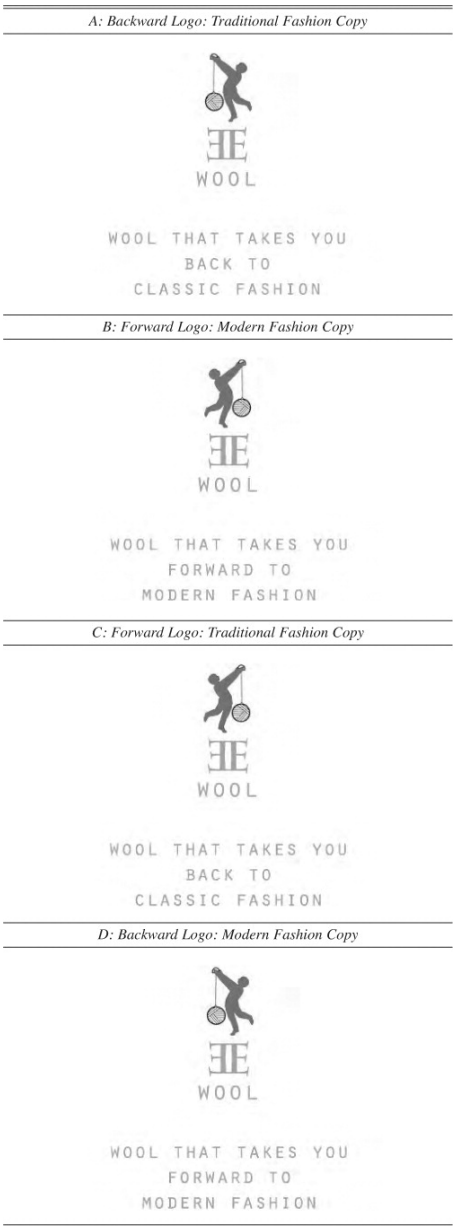

FIGURE 6. Dynamism with respect to direction of movement and company characteristic from Cian et al. (2014)

The effects on the attitude of the company, by the metaphorical match between the direction of the logo and the characteristic of the company (traditional vs modern), was studied by Cian et al. (2014). Four combinations of logos were

created as shown in figure 6. This test does not involve a test for the amount of dynamism as in previous studies but only the direction of the said dynamism. Pretest similar to that of section 2.2 were conducted and the survey group was evaluated for the attitude towards the brand. The analysis of the evaluation revealed that for a modern description, the logo with a forward direction showed the more favorable attitude towards the brand and for a traditional description, the logo with a backward direction showed a more favorable attitude towards the brand, hence proving the goal of the study.

Effect: Eye Tracking

Another study by Cian et al. (2014) was designed using eye tracking software since most of the previous results were self-reports from the people who took the survey. This was done to quantify the engagement with the pair of visuals (advertisements) in the study; one with a higher dynamism logo and one with lower dynamism logo. The engagement was quantified with the number of fixations of minimum 60 ms and the duration of the fixations. The study concluded with the result that logos with higher dynamism lead to more fixations than the ones with lower dynamism and that logos with highers dynamism lead to higher fixation duration than the ones with lower dynamism. It is stated by Cian et al. (2014) that the logo with higher dynamism receives the highest attention compared to the other elements in the advertisement.

CHAPTER III

AFFECTIVE IMAGE CLASSIFICATION

Affective Image Classification is the classification of images that affect the viewer of said images. Emotion is a stimulus in these images which is evoked in the said viewer. One of the reasons we looked at AFIC is because of its similarity with the kind of classification we wanted to do with our images of a perceived sense of movement. AFIC formed the basis of our research when we tried to establish the ability of a DCNN to learn the representation of images and classify on the basis of the representation of the said images. Hence our research is inspired by the techniques of AFIC. We will look at some existing research on the Deep Learning methods for AFIC and a general introduction to the study under AFIC.

The studies of AFIC lead us to understand the representation system of emotions and where they exist on the said system with respect to each other. This resulted in the additional study of finding out where dynamism and its absence lies on the said system with a scale of intensity called arousal and a scale of pleasure called valence as studied in Osgood (1952)

Types of Affective Image Classification

There are two types of approaches for AFIC depending on the representation of the emotions evoked by the images. The first method uses distinct categories of emotions such as joy, fear, sad to name a few and the second method plots different emotions on the valence and arousal scale.

AFIC using handcrafted features

The study in Machajdik and Hanbury (2010) mentions creating the best possible low-level features to improve the task of AFIC. To do so they exploit concepts in psychology and art theory to extract the said low-level features of images. The study was done with the goal of creating a method to retrieve images based on the affective level (emotion) compared to the then limitation of the ability to query images based on only the cognitive level. It discusses the then state-of-the-art that only consisted of a few low-level features such as the ones that were extracted using generic image processing features. Whereas the features extracted from psychology and art theory were more specific to the domain of the datasets discussed in the next section. The study also critics the generic features, arbitrary emotional categories, unpublished datasets and missing or unclear evaluations of the then state-of-the-art.

Datasets for Handcrafted Features

The study in Machajdik and Hanbury (2010) uses the existing International Affective Picture System (IAPS) dataset from Lang (2008), which has images of snakes, landscapes, puppies, babies amongst other categories. These images are labeled with their discrete emotional categories as labeled in Mikels et al. (2005). The dataset is considered as one of the standards and has featured in subsequent studies mentioned in this thesis. The second dataset is a collection of artistic photographs that are labeled by their photographers or artists who created them. The images in this dataset have been created with a conscious effort to evoke an emotion in the viewer. The third dataset consists of abstract paintings that do not contain objects as opposed to the previous two datasets. The third dataset was

added due to the absence of the effect of objects on the emotions evoked as the emotions evoked could then be attributed to features such as color and texture of the scenery instead of the objects. The images in it were labeled with emotions reported by participants of a survey.

Features Extracted from Psychology and Art Theory

The handcrafted features used for classification in the study by Machajdik and Hanbury (2010) include mean saturation and brightness of the image, pleasure, arousal, dominance values based on the saturation and brightness, and the name of the color. Some features are based on the texture of image such as the wavelet textures for saturation, hue and brightness. Other features include, the level of detail in the image, number of faces, and amount of skin in the image. The values of these features for every image are used to construct the dataset for classification.

AFIC Using Categorical Emotions

We study You et al. (2016) that largely deals with building a large scale dataset for AFIC. Since a DCNN requires a large dataset similar to ImageNet to train its millions of parameters, a custom dataset for images tagged with emotions was created. A DCNN is used to train the said dataset and compared with the methods of Machajdik and Hanbury (2010) that uses a classical machine learning approach with handcrafted features such as color and texture

Existing Datasets for AFIC

The study in You et al. (2016) suggests that even though there were some existing efforts towards building a dataset most of them were not only small but

also had a skewed distribution. The smaller size of the sets would have led to overfitting which we experience as well with the experiments that we conducted. The skewed distribution would lead to an even smaller dataset on the application of k-fold cross-validation as some categories would too few images. The existing datasets are IAPS-Subset, where the images are from Lang (2008) which are categorized into emotions by Mikels et al. (2005) and ArtPhoto from Machajdik and Hanbury (2010) which have pictures labeled and created by professional artists. Lastly, AbstractPhoto from Machajdik and Hanbury (2010) which has paintings labeled by the community. These datasets are the same as section 3.1 with names given by the study for an easier reference.

Dataset Collection

The dataset in You et al. (2016) is created by fetching 3 million images from Social Network platforms such as Flickr and Instagram termed as the weakly labeled dataset. It is labeled so because its labels are not verified by humans. This dataset is also initially skewed but this would take care of the overfitting problem as there is a large number of images per category of emotions. Images with duplicate emotion tags and duplicate images are filtered out of the set. Thereafter Amazon Mechanical Turk (AMT) is employed after a verification task for workers that included verifying the emotion of at least 10 of the 20 images designed for the verification task. After selecting 22.5% of the workers the 11,000 images per category are selected for verifying the emotions already labeled by querying Flickr/Instagram. Images whose tags were affirmed by at least 3/5 workers were added to the new strongly labeled dataset which eventually had approximately 23,000 images.

Methods for Classification

The method for learning the distribution of the 8 emotions was a DCNN using Fine-Tuning.

The category labels of emotions are classified using a DCNN from a reference architecture in Jia et al. (2014). In this method, the last layer of the DCNN pre-trained on ImageNet is removed and a layer with only 8 units instead of the 1000 units, is added. The 8 units represent the emotions in the strongly labeled dataset. The DCNN is retrained with the strongly and weakly labeled dataset with the validation done with the same dataset as they are trained with. As a baseline, a DCNN trained on ImageNet is used for feature extraction only after which Principal Component Analysis (PCA) is used to reduce the dimensionality. Finally, a Support Vector Machine (SVM) is used for classification of the features into the previously mentioned 8 emotions.

Performance of the Methods

The baseline model on the strongly labeled dataset does not perform well with an accuracy of only 32%, whereas the DCNN fine-tuned on the weakly labeled dataset has an accuracy of 46% and the same DCNN on the strongly labeled dataset has an accuracy of 58%. It is analyzed from the confusion matrix that the true negative rates were the best when fine-tuned using the strongly labeled dataset and true positive rate of fear was the highest in the baseline and not the DCNN finely tuned on the strongly labeled dataset. The feature extraction of the above DCNN models is compared with that of traditional methods of handcrafted feature extraction similar to Machajdik and Hanbury (2010) using the existing datasets mentioned in section 3.1. It is concluded that feature extraction using DCNN

performs better than the handcrafted features. The features extracted from the fine-tuned DCNN with strongly labeled dataset has the best performance, though it does perform poorly on two of the emotions on the ArtPhoto dataset.

AFIC using Emotions on the Valence-Arousal Scale

The study in Kim et al. (2018) takes a different approach to represent emotions wherein instead of using the categorical approach, it uses the Valence-Arousal scale to represent the emotions in a two-dimensional space. Instead of using a single DCNN as mentioned in the section 3.1, it uses a fusion system of different DCNNs and lower level features as a method of feature extraction and then trains it on a custom Deep Neural Network (DNN). The argument made towards this approach is that the learning process of AFIC should be more fine-grained compared to a general image classification. This is because even when the images appear similar, they can affect the viewer differently, evoking different emotions. Images that appear dissimilar could affect the viewer similarly, evoking the same emotion. The second argument is made towards bridging the affective gap between the extracted features of the static visuals and the expected emotion evoked by the person viewing the said static visual.

The suggestions made by the study include that the there is a correlation between the objects and emotion evoked by the object. Similarly, there could be a variation in the emotion evoked between objects having different backgrounds hence suggesting that the semantic information in the background correlates to the emotions evoked. The above-mentioned features are combined with low-level hand-crafted features such as color statistics to create a larger feature extraction function

which is then given as an input to a DNN for classification of the said emotions on the Valence-Arousal scale.

Creating a New Dataset

Similar to the study in You et al. (2016) the study in Kim et al. (2018) uses Flickr to fetch images that are tagged with 22 keywords including basic, prototypical emotions and affective states. About twenty thousand images are collected and then they are filtered through a process were three human subjects rated the images on their qualification of evoking emotions. After selecting images that were selected by a majority of human subjects, 6844 images were added to the dataset. This dataset was augmented by 3236 images from You et al. (2016) out of the approximately 23,000 images, thereby creating a larger dataset of 10766 images.

Then the images were labeled with their Valence-Arousal ratings represented by Self-Assessment Manikin (SAM) from Bradley and Lang (1994) using AMT. The images were tagged relative to the previous image tagged by the AMT worker so that the worker need not have the difficulty of choosing absolute ratings. The rated images obtained had a higher number of images on the Valence scale and were well distributed on the Arousal scale except for low frequencies on the extremes of the arousal scale. When compared to the IAPS dataset, the newly acquired dataset had a much better image distribution on the Valence-Arousal scale. As an additional analysis to validate the dataset. the Valence-Arousal 2Dspace is divided into four discrete subcategories including low valence, high valence, low arousal and high arousal The tags of the images are then mapped according to their emotional ratings. The resulting tags are related to the scales of the emotions across the

Valence-Arousal Scale. This concluded the validity of the dataset created for the classification task.

Feature Extraction

The study in Kim et al. (2018) suggests that the colors in an image are one of the best descriptors of emotions in an image and while it is not the only way to predict the emotion of an image it forms a good heuristic along with the other features. The first set of features is extracted by extracting the mean RGB and HSV color-space values and the quantity of the basic colors from the color histograms of the images. As proposed by Valdez and Mehrabian (1994) saturation and brightness of images are introduced as a function of Valence, Arousal, and Dominance which are then added as additional low-level features. Another suggestion comes in the form of local feature extraction which includes GIST descriptor from Borth et al. (2013) for detecting scenes and a local binary pattern descriptor for detecting textures. These set of features are not as extensive as in section 3.1 but are complemented with object and scene detection features.

The objects in the feature is another important feature to predict emotions. This suggestion is backed by an experiment where the tags containing the name of the object in the IAPS dataset is mapped to the valence-arousal score from the word emotion dictionary from Warriner et al. (2013) which has associated valence-arousal ratings for words. The valence-arousal ratings of the words are mapped to the valence-arousal ratings of the images in IAPS and it is found that there is a high correlation between the emotion evoked by the object then that by the image itself. On the basis of this conclusion, a DCNN pre-trained on the ImageNet

dataset is used to predict the object. The output of the final layer is used as a feature vector.

Another feature included the semantic features from the background which the study in Kim et al. (2018) claims has the ability to part take in emotion prediction. This is achieved by using a DCNN as mentioned in Wu et al. (2016) which does semantic segmentation of the pixels in the image to 150 semantic categories which are used as a feature vector in addition to the previously mentioned vectors.

Emotion Prediction Model and its Performance

The model that is used to train the above-mentioned feature vectors is a DCNN with an input, output, and three hidden layers. It uses Stochastic Gradient Descent (SGD) for optimization and Mean Squared Error (MSE) as a loss function. The output of the final layer predicts the valence or arousal ratings.

The performance of the model was judged by comparing the 3 models namely AlexNet from Krizhevsky et al. (2012), VGG16 from Simonyan and Zisserman (2014) and ResNet from He et al. (2016), Targ et al. (2016), and Deng et al. (2009b) used for feature extraction of object categories and additional category-level features from Yu et al. (2013). VGG16 performed the best in terms of valence and arousal. The performance of the features was judged by training every feature type with a similar model as proposed for the emotion prediction. The object detection feature extracted from VGG16 performed best for valence, and from AlexNet for arousal. Expectedly, the lower-level features performed the worst.

Comparing with Transfer Learning

The study in Kim et al. (2018) compares the performance of their emotion prediction model with that of a pre-trained AlexNet and VGG19 from Simonyan and Zisserman (2014). Most of the hyperparameters are kept constant but the Transfer Learning is done using two methods. In the first method called frozen, all the convolutional layers are frozen and only the fully-connected layer is allowed to be trained. By doing so, they make sure that the lower and higher level features learned from the pre-training are not changed thereby not changing the feature extraction and only allowing the classifier to change. In the second method termed train, the entire CNN is allowed to train. The final layer is a single unit kept consistent across all the models. From the experiments, it was concluded that the second method train performs better than the first method as the loss is the lowest. The pre-trained VGG19 performs much better than the AlexNet with the train method. When compared with a Linear Regression Model and Support Vector Regression they both outperformed the AlexNet and VGG19 models but underperformed compared to the emotion prediction model.

CHAPTER IV

TRANSFER LEARNING

Given the unavailability of a large-scale dataset for our problem defined in chapter I, and studying the successful use of transfer learned DCNNs with AFIC, Transfer Learning was considered as a probable solution for our problem as well. This warranted a study into Transfer Learning so that we could devise methodologies and hyperparameters for our own experiments.

According to the study in Oquab et al. (2014), it is acknowledged that DCNNs perform better than traditional algorithms on image classification specifically object recognition tasks. This is largely due to the feature extraction process of DCNNs which performs better than feature extractors such as SIFT from Lowe (2004) and HOG from Dalal and Triggs (2005). Graphics Processing Units (GPUs) have enabled faster training due to the embarrassingly parallel matrix computations which are the fundamental computations of a DCNN. This scalable computing ability has made training millions of parameters of a DCNN on large-scale datasets feasible. Given that GPUs are easily available and scalable the performance capability of a DCNN is largely attributed to the availability of large datasets such as ImageNet, Caltech256 from Griffin et al. (2007), Pascal VOC from Everingham et al. (2010). It is argued in Oquab et al. (2014) that it is infeasible to construct an ImageNet scale dataset for every image classification or object recognition task. Hence the need for methods to transfer knowledge from a task in one domain to another task in the same domain or to another task in a different domain.

Tasks in the Same Domain

The databases mentioned in the above section have differences between them since they were collected by different people. Datasets such as Caltech256 and ImageNet have categories of images where the object referencing the category label is centered whereas in datasets such as Pascal VOC images have more spatial invariance in terms of different backgrounds or positioning hence affecting the performance of the DCNN they are trained on. Since Pascal VOC is a much smaller but a different dataset compared to Caltech256 and ImageNet, transfer learning is considered as a solution to mitigate the problem of the dataset size and in an effort to improve object recognition performance with Pascal VOC.

The Model Architecture

The base DCNN architecture in Oquab et al. (2014) has 5 convolutional layers along with 3 fully-connected layers. The model is modified by replacing the last fully-connected layer with two other fully-connected layers. The last layer of the modified model is remapped to represent the categories of the Pascal VOC dataset for the object classification task. The weights of all the convolutional layers and the first fully-connected layer are unchanged in the modified model. The key idea is to train the base DCNN on a large-scale dataset such as ImageNet, and then use the weights of the base model to act as mid-level feature extractor for the modified model.

Training and Classification Strategy

Images in ImageNet are preprocessed and the object to be recognized in the images is in the center whereas in the Pascal VOC dataset images have objects

situated in a scene with varying backgrounds and with multiple objects in the same image. This adds what is called a database capture bias as mentioned in the study by Yosinski et al. (2014). To address this difference in the datasets, the independent training of the last two fully connected layers is done using a sliding window object detector.

As mentioned in Oquab et al. (2014), the sliding window object detector method consists of creating patches of various scales extracted from the images to be trained. The patches are then labeled by what they contain which could be a partial or entire object and/or just the backgrounds. The labeling is done by comparing the area of the bounding boxes of the patch and the ones in the original image. The qualification for a label is based on two thresholds, for a complete match or a partial match with the condition of no-overlap between more than two objects. The unqualified patches are tagged as background which are then resampled to create a more balanced training set. This method brings the dataset closer to the kind in ImageNet in terms of having the object in the center. These patches are then used to train the model architecture. A separate set of patches are used for testing in which they are scored by the confidence of the class the patch belongs to. The score is adjusted to classify patches that have a higher confidence.

Transfer Learning Experiments

The base model is trained and an adaptive learning rate is used where its rate is reduced until the loss function for training is stabilized. The model is trained on the previously mentioned ImageNet with over a million images across thousands of categories. The training on the then available hardware took about a week, and an 18% top-5 error rate was achieved. After transferring the weights from

the base model to the modified model, it is retrained with the Pascal VOC 2007 dataset which achieves a better score per class accuracy compared to the then pre-existing models, for example, the 2007 image recognition challenge winners. The modified model retrained on the Pascal VOC 2012 dataset does not perform as well as compared to the pre-existing models, only outperforming them in a few classes.

In the effort to determine the effect of the overlap of categories across the ImageNet and Pascal VOC datasets, the base model is pre-trained on two different sets of the ImageNet categories. As a baseline, the base model is trained with random 1000 categories of images. The weights are transferred to the modified model and the per class accuracy on the retrained Pascal VOC 2012 dataset is reduced. The same modified model when pre-trained on an augmented set of categories containing the random 1000 and overlapping categories, is retrained on the Pascal VOC 2012 dataset, outperforms the baseline and the winner of the competition for which the dataset was created. Thereby concluding that there was a benefit from the overlapping categories.

The size of the final layers was fine-tuned for the modified model. It was observed that removing or adding a layer, resulted in a marginal drop and increase of 1% of the accuracy respectively. Thereby concluding that the selected number of finals layers was appropriate for the task.

Tasks in a Different Domain

To demonstrate the ability to learn tasks in a different domain, the pre-trained classifier in Oquab et al. (2014) was used to retrain on the task of action recognition in the Pascal VOC dataset. This dataset consisted of images of humans performing a certain action or interacting with an object. The said classifier was

able to outperform the model used without any pre-training. They were able to achieve better performance when the final fully connected layers were allowed to retrain. We have also seen from our previous study in AFIC from chapter three that it is possible to transfer knowledge from a task in one domain to a task in another domain.

Transferability of Layers

Given that we now know the benefits of Transfer Learning by empirically understanding its performance differential compared to training from random initialization of neurons, we look at a study concerning the transferability of features across tasks. A typical DCNN contains a series of layers across which weights are adjusted using a back-propagation algorithm. The layers with their weights act as feature extractors for the network which in turn act as input to the successive layers. Every layer extracts different features with increasing complexity. In terms of transfer learning, the lower-level features of a DCNN trained on datasets such as ImageNet extract generic features. The higher-level features tend to more specific to the dataset with which we retrain the DCNN. As we have learned from our studies in AFIC, given a small dataset in our domain, transfer learning is more likely to not overfit compared to random weight initialization. Hence we look at the transferability of this layers in order to understand, how many layers we can transfer or fine-tune.

As mentioned in the study by Yosinski et al. (2014), the lower layers of a DCNN extract more generic features that are common to Gabor Filters as opposed to the higher layers. The last layer of a DCNN is specific to the categories or classes of the dataset with which we retrain the DCNN. The study suggests that

the lower-level features are common to all DCNNs trained on a natural image dataset such as ImageNet irrespective of the type of classification supervised or unsupervised. It would be safe to assume that the networks we train in our experiments would have learned the same generic features as well, in their lower layers. Given the concept of lower layer being generic and higher layer being specific the study by Yosinski et al. (2014), explores the degree of the said concept and the layer where the split between the two kinds of layers occurs in a DCNN

The study suggests two methods of performing the transfer of knowledge from ImageNet to another domain. The first method, generally referred to as frozen involves the transfer of weights from a DCNN trained on ImageNet for some x layers, where x is a hyperparameter. Here x is less than n where n is the total number of layers in the DCNN. After transferring x layers the rest (x-n) layers are trained on the new dataset. The x layers which are frozen act only as a feature extractor and do not adjust their weights in backpropagation when the (x-n) layers are trained. The second method is referred to as fine-tuning in which the previously mentioned x layers are allowed to adjust their weights in backpropagation when the (x-n) layers are trained. The study suggests that the one way to decide which method to use is a function of the size of the new dataset to be trained and the number of parameters of the architecture. If the new dataset is small and the number of parameters is large then fine tuning might overfit the network in which case freezing the layers is preferred but if the new dataset size is large and the no.of parameters are relatively small then the x layers can be fine-tuned. These suggestions contain many hyperparameters and the only way to determine them is empirically, as we do in our experiments.

Degree of Generality

The study by Yosinski et al. (2014) experiments with the generalizability of layers in a DCNN. Two tasks are created, namely A and B such that A and B each have half of the images in ImageNet. Since ImageNet has categories which have similar images such as a different breed of an animal, both the datasets have some similarity in their images. Two replicated DCNNs of eight layers are trained with each of the dataset named baseA and baseB respectively. Multiple networks called transfer learned networks are created using the naming scheme (dataset A)-n-(dataset B) where n defines the layer number until which the layers are frozen from the input. Layer 0 to Layer n have weights transferred from dataset A and Layer (n+1) to the layer before the final layer are trained on the dataset B with randomly initialized weights. Networks that have weights transferred from the same dataset on either side are referred to as the control network. The same study is conducted by fine-tuning the layers instead of just freezing the layers. These studies are conducted for all layers. The reason behind the study was that if the network (dataset A)-n-(dataset B) performs as well as the control network (dataset B)-n-(dataset B) then the layer nl is generic or else it is specific. According to the study in Yosinski et al. (2014), fine-tuning reveals the fragile co-adaptations of neighboring neurons. Co-adaptation, in this case, means the ability of neighboring neurons to efficiently transform features (or any such interaction) from one layer to the other.

The accuracy of the baseline network baseB is at 0.62. The control network is trained using the freezing method. When the initial or last layers of the control

network are spliced [1], the neurons are able to co-adapt to the different weights just as in the base network. Splicing of the middle layers leads to a performance drop. This effect is mitigated by fine-tuning as the fine-tuned control network performs similarly as the base network. The transfer learned network, trained using freezing, initially performs as good as the base network. Though, when the higher layers of the network are spliced, the performance drops due to the fragile co-adaptation of the neurons in the layers that were transferred from the base dataset. As the spliced layer level is increased, the network fails to generalize as the lower layers have transformed from generic layers to layers specific to the base dataset. The transfer learned network, on fine-tuning, shows an improved performance compared to the base network. The fragility of the co-adaptations of neurons is intact or strengthened since the transferred layers are also allowed to retrain. This shows that fine-tuning can be used instead of randomly initializing weights when both the datasets are similar to some degree.

A study in Yosinski et al. (2014) shows a drop in accuracy as the level of the spliced layers is increased for a transfer learned network trained on dissimilar datasets. This behavior occurs as the network is unable to generalize because the weights are transferred from dissimilar datasets. Since the images are different in the dissimilar datasets, the features learned by the network could be distinct depending on the images. Networks with randomly initialized weights tend to perform worse compared to the transfer learned networks trained using dissimilar datasets. Even though it might be possible to find the optimal randomly initialized weights to train the network, the weights of the transfer learned network are a better heuristic.

[1]Spliced: The word spliced is a synonym for the word attach and in the context of the thesis it is used to denote the layer which differentiates the weights of the two datasets

CHAPTER V

NETWORK ARCHITECTURE AND AUGMENTATION

Network Architectures

Introduction

In an effort to use DCNNs for the problem defined in chapter I, we study their architecture. The reason we use reference architectures is so that we have a heuristic of its performance on standard datasets and some of its hyperparameters. Towards this, we select two popular architectures namely VGG16 and Inception V3 from Szegedy et al. (2016). Both these architectures have deep convolutional layers that have shown benchmark performance on various standard datasets. We discuss the architectures of both the models and some of its differences.

VGG16

The VGG16 architecture first appeared in the study by Simonyan and Zisserman (2014) and it was built in an effort to improve upon the existing eight layer architecture from a study in Krizhevsky et al. (2012), which won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012. The architecture mainly deals with exploring the performance effect on standard datasets such as ImageNet by increasing the depth of the layers. This architecture turned out to very successful after its victory in the ILSVRC held in 2014 in the localization and object classification tasks. It is considered as one of the benchmark architectures in computer vision tasks. The architecture was designed by the Visual Geometry Group (VGG) at the University of Oxford and the 16 its name stands

for the number of layers in the architecture. The 16 layer architecture used in our experiments is given in figure 7

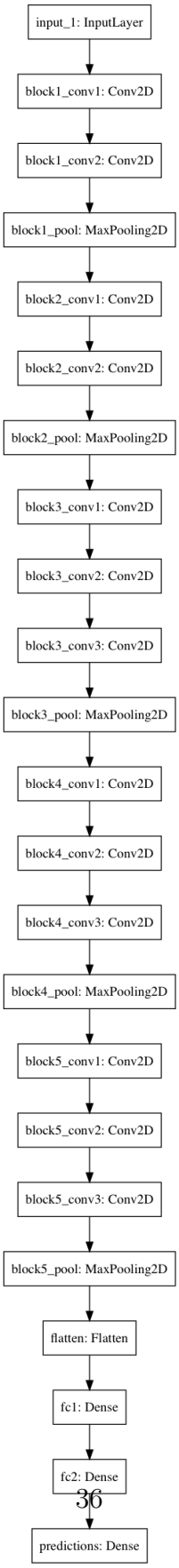

FIGURE 7. Architecture of VGG16

As seen in figure 7, the first layer called the input layer takes a fixed input of 224x224xn pixels where the first two values are the width and the height of the images fed to the network and n is the number of channels where n could be 1 (for grayscale) or 3 (for color or RGB images). This network normalizes the input by subtracting every pixel with the mean pixel value of the image. Next in the architecture are the convolutional layers and the final classification layers. The convolutional layers act as feature extractors and the classification layers learn to classify the features of images given as inputs. The convolutional layer consists of convolutional blocks which has trainable filters of the size 3x3. The size 3x3 was selected in the study by Simonyan and Zisserman (2014) arguing that that is the smallest size of a filter that is able to capture the notion of the four directions on the image. These filters are moved across the images with a stride 1 in this architecture and for every move the filters convolute the image. The filters have learned the values which are then multiplied by the pixel values the filter mask's over, to generate a convoluted map of that patch of the image. These patches are padded by 1 pixel to maintain the spatial resolutions of the image patch. These maps generated by the filters are then pooled by the max-pooling layers where the maximum value of a 2x2 pixel window is computed to highlight the most significant value of that map. These windows are moved over a stride of 2 pixels in this architecture. As the images is convoluted throughout the convolutional layers the size of the features decreases whereas the size of the channel increases. Such convolutional layers are stacked together and some are followed by the max-pooling layers. This stacking forms the integral part of the architecture as it determines the features extracted from the image. The lower layer features are known to be more general whereas the higher layer features are known to extract higher-level features

which are more specific to the images. Followed by the convolutional layer is the classification layer which contains two fully-connected layers which has all neurons connected to all neurons in the successive layer unlike in the convolutional layers. In this architecture the two fully connected layers have 4096 neurons each with the final layer having 1000 units for ImageNet classes or as suited for the trained application. The final layers have softmax activation which generates the class probabilities while the fully connected layers have Rectified Linear Unit (ReLu) activation unit as used in Krizhevsky et al. (2012).

Inception V3

The creation of networks such as AlexNet and VGG16 whilst proving their performance at ILSVRC competitions also spurn research in the architectures of the DCNN. According to the study in Szegedy et al. (2016), the direct performance improvement with these networks led to improvements in real-world applications. Though there was still a need for network design that was able to train equal or fewer parameters thereby offering computational efficiency with equal or better performance. VGG16 or AlexNet offer structural simplicity so that they are easy to design, understand and modify. We came across this same problem with our experiments where it was easier to decide how to transfer knowledge from VGG16 than it was from Inception V3. Though we still included Inception V3 in our experiments as the study in Szegedy et al. (2016) states that it performs better in image classification task when compared to AlexNet or VGG16. The biggest potential performance boost was offered by the marginally less number of parameters required by Inception V3 to perform better than AlexNet or VGG16 when it came to image classification performance. AlexNet has approximately

63 million parameters and VGG16 has 134 million parameters whereas Inception V3 has only 21 million parameters a third of AlexNet and a sixth of VGG16 while offering better performance. This computational efficiency allows the use of such networks in constrained environments such as mobile operating systems. As mentioned in the chapter I we wanted the eventual accessibility of the perception of the sense of movement we implemented Inception V3 in our experiments.

This optimization in the number of parameters is achieved by using the Inception Module mentioned in GoogleNet from Szegedy et al. (2014) and then applying certain optimization techniques such as Factorized Convolutional and Aggressive Regularization explained in the study Szegedy et al. (2016) to scale up the original network in an efficient way. It is argued that naively increasing the number of layers or filter sizes will only negatively affect the number of parameters and computational efficiency due to the inflexible structure of GoogleNet. We will not be describing the structure of the inception modules and/or the optimization offered by Inception V3 as they are better explained by Szegedy et al. (2014) and Szegedy et al. (2016) respectively. The actual figure of an Inception V3 is not included for spatial constraints of the thesis. Instead, a brief summary of the network is provided in table 1 where the Inception modules are abstracted. The rest of the network contains layers similar to that of VGG16 mentioned in the previous section with the difference that only one convolutional layer is padded in the non-Inception layers along with the different size of the input image in the input layer.

[1]The Inception modules have variable patches as mentioned in Szegedy et al. (2016)

Type	Patch Size/Stride	Input Size
conv	3*3/2	299*299*3
conv	3*3/1	149*149*32
conv padded	3*3/1	147*147*32
pool	3*3/2	147*147*64
conv	3*3/1	73*73*64
conv	3*3/2	71*71*80
conv	3*3/1	35*35*192
3*Inception	variable [1]	35*35*288
5*Inception	variable	17*17*768
2*Inception	variable	8*8*1280
pool	8 * 8	8 * 8 * 2048
linear	logits	1 * 1 * 2048
softmax	classifier	1 * 1 * 1000

TABLE 1. Inception V3 architecture recreated from Szegedy et al. (2016)

DCNN for Augmentation

We use a standard architecture from Chollet et al. (2015) in our data augmentation experiments. The architecture consists of 3 convolutional layers with 3x3 filters, ReLu as its activation function, and a max pooling layer in every convolutional block with a pool size of 2x2. This network is similar to AlexNet but has fewer convolutional blocks. The output of the convolutional layer is given as an input to the fully-connected layer of 64 units. The last layer has a single output unit as we use a sigmoid activation function for our binary classification task.

Augmentation

DCNNs have a deep architecture that allows millions of parameters to be trained but this feature can be exploited only if there is an equivalent amount of Big Data to train it on which is not often the case. Even though now we have large datasets such as ImageNet and Caltech-256 from Griffin et al. (2007), these

images are constrained by their domain and there still is a lack of data in other domains without which we can not take advantage of the deep architectures of a DCNN. This lack of availability of data leads to overfitting of the neural network as it is not able to generalize well enough to be used on test images. There are other techniques such as dropout and batch normalization which have shown to reduce overfitting as shown in the study by Krizhevsky et al. (2012) and data augmentation has also found itself in image preprocessing pipelines.

Image Augmentation is a technique of augmenting or amplifying image data from existing images such that a DCNN is able to learn the features of the image without adding image biases, for example, classifying bananas that are only right tilted because that is how they appear in the dataset. These techniques are functionally efficient as they are label preserving and do not transform images into something that they are not. They improve the training performance of said models as well as the testing performance.

The technique of Image Augmentation, that has shown benefits in DCNN, has been limited to few basic techniques such as translations and rotations. Most of the studies do not delve into the reasoning for the use of particular techniques, how they are selected and or the hyper-parameter selection related to them. Hence we study other techniques that might be useful for a given task and the technique or combination of techniques that work for the given task. We would be tackling the former question to find other techniques in image processing to augment the data.

Related Work

Even though data augmentation techniques have been used before in machine learning models, deep learning is where they have shown considerable performance improvement due to the scale of the very task.

One could attribute the popularity of data augmentation to a DCNN constructed by the study in Krizhevsky et al. (2012) due to its citation for data augmentation techniques in other studies in the literature. Even though the work is focused on the architecture construction such as using overlapping pooling and local response normalization a considerable effort was spent on reducing overfitting due to its 60 million parameters. Other than dropout a technique that drops some neuron outputs randomly, image data augmentation without which the network suffers substantial overfitting as stated in the study by Krizhevsky et al. (2012) was the primary technique to reduce overfitting. Translations and horizontal reflections are applied by extracting 224x224 patches from their 256x256 images which augment the images by 2048x. Another technique was altering the intensities of the RGB channels in training images to capture an important property of natural images that color variations do not change the labels of an image. This particular technique reduced the top-1 error rate on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by a 1% though no improvement was measured by the first technique.

The study of DCNNs used with ImageNet spawned many studies that target a performance improvement over a previous system to compete in the ILSVRC. One such study by Howard (2013) focuses on improving the translations and rotations for data augmentation. It improves translational invariance by changing the scaling and cropping images in such a way that object features are

not sacrificed during rescaling. It also improves lighting invariance by randomizing contrast, brightness, and color instead of just adding random lightning noise similar to the systems before it. As mentioned before the augmentation techniques help training data as well as. The testing images are augmented by doing a join prediction on greedily selected predictions using three different scales and 10-15 other subsets of transformation instead of combining all predictions. The study in Howard (2013) also deduces that compared to Krizhevsky et al. (2012), the augmentation improves the error rate but adding another fully connected layer does not improve the rate. Its greedy prediction selection also improves run time while reducing the error rate compared to the same baseline, thereby attributing the improvement solely to the augmentation.

Augmentation Techniques

These are some of the techniques that are regularly used and some additional techniques from image processing techniques from imgaug [2]

Rescaling

Images are rescaled so that the model can learn positional and size invariance of the object as seen in figure 8.

[2]imgaug is an image transformations library built in python. The images in the following section are used from the demonstration images from the library at https://github.com/aleju/imgaug

FIGURE 8. Example of Rescaling an Image

Translations

The images are translated so that the model learns to recognize the parts of an object as a part of the object with labeled confidence as seen in figure 9

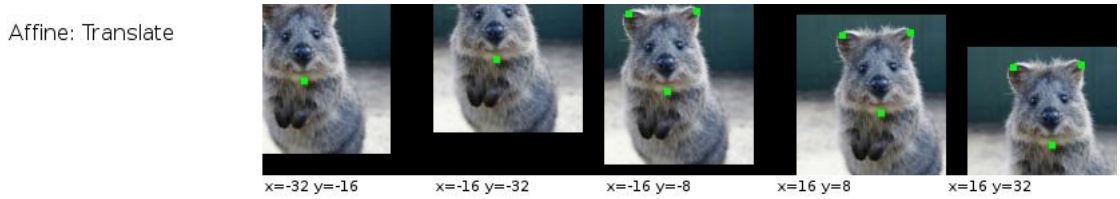

FIGURE 9. Example of Translating an Image

Gaussian Blur

It is a technique of blurring an image in image processing by using a Gaussian function. It is used to reduce image noise and detail and to enhance image structures in computer vision. Example of the technique is given in figure 10

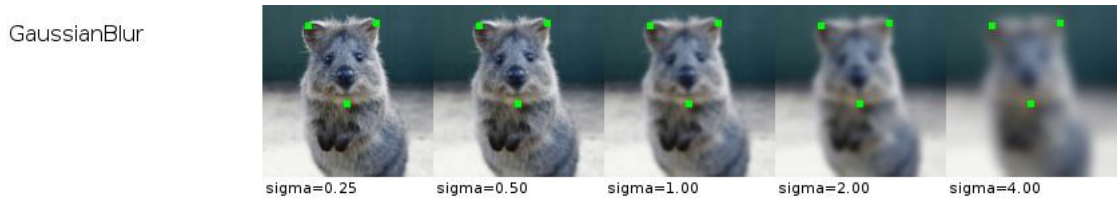

FIGURE 10. Example of adding Gaussian Blur to an Image

Elastic Transformation

It is an image transformation technique in which certain pixels are moved around locally to create an elastic distortion as seen in figure 11

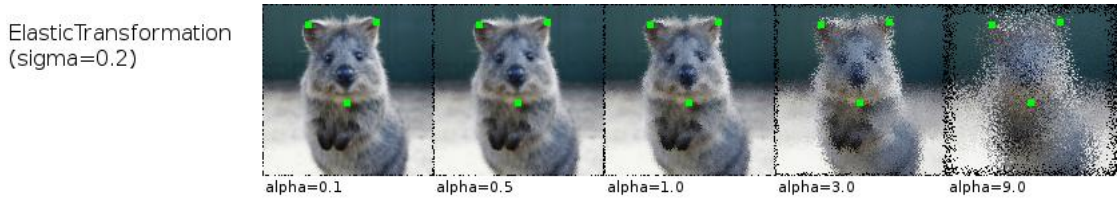

FIGURE 11. Example of adding Elastic Transformations to an Image

Dropout

It is a technique in which certain fraction of pixels or squares (coarse) are set to zero or dropped as seen in figure 12. This technique is also used as a regularization technique to reduce overfitting in DCNNs as it is applied in the feature space by randomly turning some neurons off. We look into its applications in the feature and the data space by applying to input images as seen in figure 12

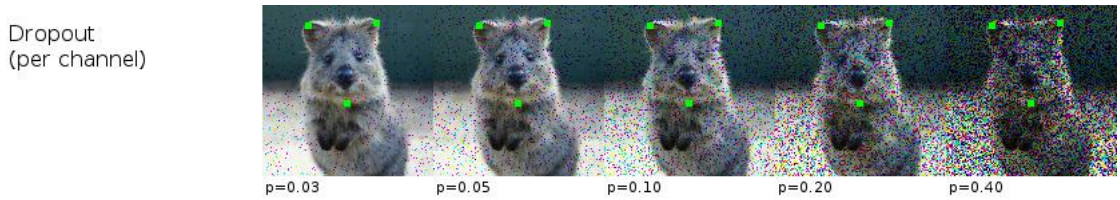

FIGURE 12. Example of adding data-space Dropout to an Image

CHAPTER VI

DATASETS, APPROACH, AND EXPERIMENTS

Image Collection for Our Dataset

To solve the problem of the perceived sense of movement in static visuals our first intention was to find a pre-existing dataset of images. Though due to the very novelty of the task were unable to do so and hence we had to build our own dataset from scratch. We created our own dataset which was manually labeled by us but collecting a large amount to train our DCNNs was infeasible as we did see severe overfitting. Hence we chose to fetch the images using large image repositories such as Flickr. We chose Flickr as the repository to download images from since its a repository where people in the world upload their images and give it their own tags. We were interested in these tags as these tags tend to be rather specific, in terms of what the photos contain. Going into the collection of images, we knew that the dataset we would collect would be weakly labeled as it would contain noise to a certain extent. We use the adjective weakly labeled as the tags of the images are not verified by humans and hence some tags could be disputed. Though the images from Flickr would also help our DCNN to generalize due to the availability of a large number of images. The collection of these images was made easier by the Application Programming Interfaces (APIs) provided by Flickr.

To collect the images we modified the pre-existing Python scripts [1] using Flickr's APIs which fetches images by keywords. We chose the generic keyword dynamism and not dynamic as the images return by dynamic seemed to have

[1]The pre-existing Python scripts were used and modified from https://github.com/bertcarremans/Vlindervinder

a lot more unrelated images than we expected. Querying Dynamism returned images that had implied movement in the images. Since we intended to do binary classification we used the keyword still to find images that had stationery objects. This created a weakly labeled dataset as it also contained noise in terms of incorrect classifications. Even though the keyword still returned millions of images which would have added a lot to our training phase and the eventual performance of the DCNN but we wanted to balance with the distribution of the images with dynamism. Since there were only a few thousand images related to dynamism we fetched an equally balanced set which also led to a faster training of our DCNNs.

We fetched about three thousand images with dynamism and three thousand images that have an object that was still. The images were fetched by sorting with relevance on Flickr, and with the best size available starting from the original size. Given below in figure 13 and 14 are some examples of the dynamic images and figure 15 and 16 are the example of images with 'still' objects in them

FIGURE 13. Dynamic image from our dataset

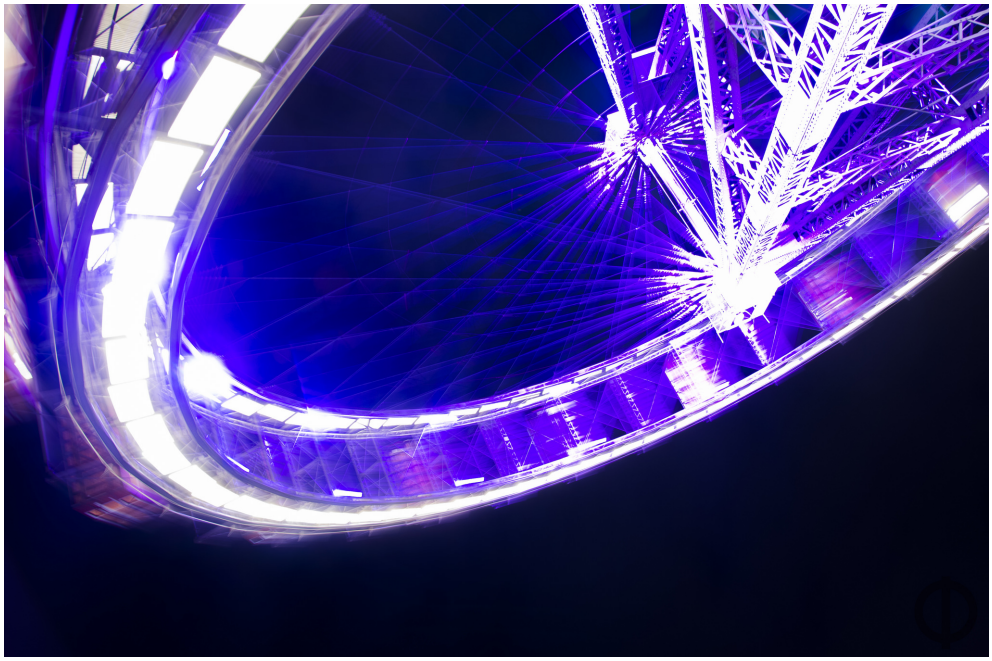

FIGURE 14. Dynamic image from our dataset

FIGURE 15. Still image from our dataset

FIGURE 16. Still image from our dataset

IAPS and OASIS Datasets

IAPS - International Affective Picture System

The next dataset we needed, was to assess if the perceived sense of movement or dynamism had an effect on emotion. We found the IAPS to be the standard dataset used in Machajdik and Hanbury (2010), You et al. (2016), and Mikels et al. (2005) amongst other studies, discussed in the chapter III of this thesis.

IAPS has 1183 images that are tagged on the emotional scale of valence and arousal. Valence is the scale that ranges from unpleasant to pleasant and Arousal is the scale that ranges from calm to excited. The images are tagged using the Self-Assessment-Manikin from Bradley and Lang (1994) that has a figure representation of the ranges of emotion on both the scales. The images are tagged by taking a mean of the ratings, as reported by a survey group, on the emotional scale.

OASIS - Open Affective Standardized Image Set

To compare our DCNN with different distributions of images tagged on the emotional scale. We retrieved the OASIS dataset from Kurdi et al. (2017), which contains 900 images similar to IAPS and is tagged with a similar valence-arousal scale. The difference between the two datasets is that OASIS is more freely available and contains recent images tagged by a group of people online. This dataset contains broader categories of images compared to IAPS. We use this dataset for the same experiment as IAPS.

Our Approach

We discuss our approach to solving the problem of classifying images with a perceived sense of movement or dynamism. Our approach is inspired from the understanding we have gained from the previous chapters covering the various topics. The second chapter introduces our problem, its definitions, degrees, types, and effects in the real world. To solve the said problem we look at AFIC as it is the closest to the domain of our problem. Our methods of data collection are inspired from the study in You et al. (2016), and Transfer Learning as a method for our approach from the study in Kim et al. (2018), both studied in the chapter III. We use the concept of emotion representation on the valence-arousal scale from the study in chapter III to find if there is a correlation between the emotions in an image, and dynamism. Establishing a correlation could potentially give static visual creators such as artists, to vary the dynamism in an image to adjust the emotion evoked by the image. We then delve deeper into an empirical study of the theory of Transfer Learning along with its methods to transfer knowledge from another domain to that of ours. In doing so we experiment with the different

ways we could perform transfer learning to solve our said problem. Towards our approach, we study the architecture of the networks we intend to use and study the transformation of images to introduce spatial invariance. We augment our novel dataset to compensate for its relatively smaller size compared to ImageNet. This is followed by a brief introduction to the datasets we use in our experiments. We discuss the pipeline of our experiments, the results, and observations of the said experiments.

Experiment Pipeline

We begin our series of experiments by selecting the kind of image transformations that we could apply to augment our dataset. We create a baseline by applying no transformations and then gradually try a different kind of transformations to see which one minimizes the validation loss. Once we have a set of transformations that are optimal, we use them as constants in all our experiments.

We then train our image dataset on multiple networks to observe which returns the best result. We start with a basic DCNN and reuse the one we used for the augmentation experiment. We decide to use the VGG16 architecture for its structural simplicity and the Inception V3 for its performance optimizations over networks similar to that of VGG16. We train a VGG16 that is randomly initialized without Transfer Learning to see if our dataset with augmentations is enough to learn the intended classification task. Thereafter we train the VGG16 and Inception V3 using Transfer Learning.

We perform the Transfer Learning experiments in two separate ways. As the first method, we use the frozen method in which we do not transfer all the weights

from a network trained on ImageNet, but only the weights of the convolutional layers are transferred to our network. The rest of the remaining network is essentially the fully-connected classifiers which are only initialized with random weights. Our dataset is then retrained on the network where the frozen layers due to their inability to learn then act as feature extractors. In the second method called fine-tune' we modify and use the fine-tuning technique. In this technique unlike the previous one, we transfer all the weights from a network pre-trained on ImageNet to our network. This is done so because as mentioned in section 4.3 the weights of a DCNN are better initialized with weights of another DCNN from a dissimilar domain than randomly initialized weights. We then fine-tune the layers of the DCNN and also find the optimal layer to fine-tune until as to maximize the performance of the network. All the above-mentioned networks are validated from the images in our dataset but are different from the ones in training.

Once we have validated our networks, we select the best performing network to be used in our experiment to observe the correlation between dynamism and emotions. We perform this experiment by using our best performing network to classify the images in IAPS and OASIS based on their dynamism or its absence. We then plot the emotions of the images of IAPS and OASIS on the valence and arousal scale and then visualize the classifications to observe a relationship between emotion and dynamism.

Experiment Constants and Environment

In the following experiments, we make certain selections and choices that we discuss in this particular section. Since we use Transfer Learning to transfer knowledge from another domain to solve our problem, we choose ImageNet as the

ideal dataset to transfer features. ImageNet has the highest probability of having images similar to the ones we obtain from Flickr. Just as seen in the section 4.3, ImageNet has shown the ability to add generic features to DCNN. Hence we have used ImageNet to pre-train our DCNNs in the experiments.

The following experiments were programmed using Keras a popular framework for deep learning that uses TensorFlow for tensor or matrix computations. Keras supports a flexible and relatively easy way to parallelize the experiments across Graphical Processing Units (GPUs). These experiments were performed on the Talapas Supercomputer at the University of Oregon. Most of the experiments are computationally expensive due to the size of the datasets and the use of DCNNs. Every experiment was parallelized across four Nvidia K80 GPUs along with a 28-core Intel CPU which made the experiments feasible.

In the following experiments, we keep the distribution of the training and validation set constant where 70% of the dataset about 4200 images is used for training and 30% or 1800 images are used for validation. The images are fed to the network using a generator that yields batches of 100 images. We use a generator here to avoid any out of memory exceptions as the size of the dataset is at 6000x244x244 not accounting for the augmentations we have used. Most of the experiments take about an average of 19 hours to finish when parallelized across four GPUs.

Experiment Results and Observations

Transformations for Augmentation

In this experiment, we attempt to find a good heuristic of transformations that we could apply to images, such that it maximizes the accuracy of the dataset

trained on a DCNN. Since our collected dataset takes about 19 hours to train we use a pre-existing dataset with a known performance so that we can tune the types of transformations we would apply to our images. We use the dataset with images of cats and dogs from CIFAR-10 obtained from Keras. It has 1500 images of cats and 1500 images of dogs. For the purpose of this experiment, we split the dataset into 70% for training and 30% for validation of the DCNN. We use the standard architecture mentioned in section 5.1 that is able to learn the classification of cats and dogs to an accuracy of approximately 0.81%. The model is compiled using RMSProp as its optimizer for back-propagation and binary-cross entropy as its loss function. The images in the validation and test phase were only rescaled as mentioned in section 5.2.

Without Augmentation

In this experiment, a baseline was constructed using no image transformations, we only rescaled the images. A dropout of 0.5 was used in the final layer to reduce the expected overfitting. It can be seen from the figure 17 below that network converges well on the training data but ultimately the model overfits even after using dropout. This is the result of having very fewer images such that the network does not generalize over the dataset.

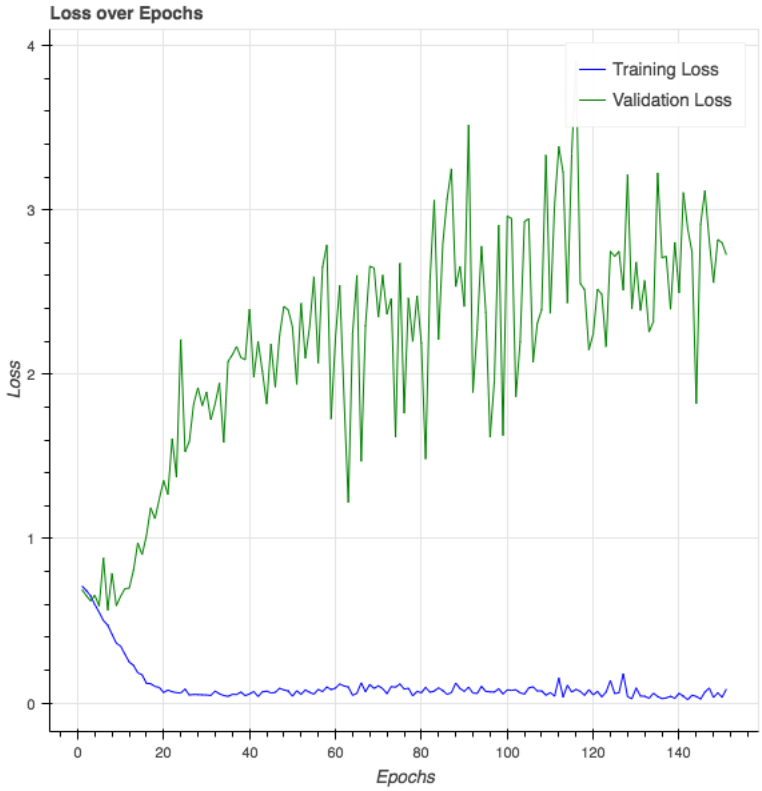

FIGURE 17. Baseline Loss

With Augmentation

In the next experiment, we tested image transformations such as Elastic Transformations described in section 5.2 where the network did converge but validates to a low accuracy of 0.69% as the transformations were uniformly applied. When only random images were elastically transformed and contrast normalized the network validated to an accuracy of 0.75% which was only marginally better. We applied dropout in data-space, feature-space, and in both, but the best performance was obtained in the feature-space. We experimented with tuning the parameter (gaussian blur sigma) of gaussian blur and even though we obtained an optimal parameter, the accuracy of the network with the optimal parameter was only at 0.76%. The best results were obtained when resizing was combined with

horizontal flips, random zooms, rotations, width-height shifts, and dropout where the network converged and validated with an accuracy of 0.80 though it had the lowest validation loss of 0.56 as seen in 18.

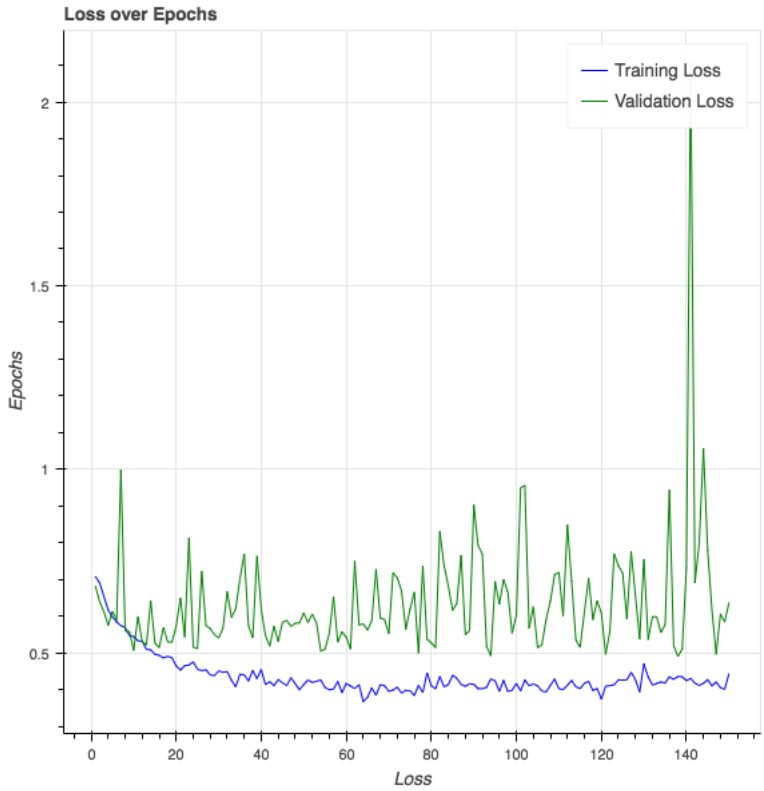

FIGURE 18. Network with Augmentation

Observation

In the above experiments, we could not find the best singular transformation to apply that could have improved the performance of the network. Even though it may exist, it may not be computationally feasible to find the optimal transformation or the optimal parameters for the transformation as the benefits are very little. In the next set of experiments, we use the best performing transformations in the above experiment that help augment our dataset and introduce spatial invariance of the objects in the images.

Baseline DCNN

We continue to use the same network architecture as the one used in section 6.4 and described in section 5.1. We experiment without using Transfer Learning as a baseline to our other network architectures. It can be seen in the figures 19 and 20 below that the network does not converge after 50 epochs as the training loss is down to 0.53 but the accuracy is only at 0.73. The validation loss is down to 0.57 and accuracy only at 0.73.

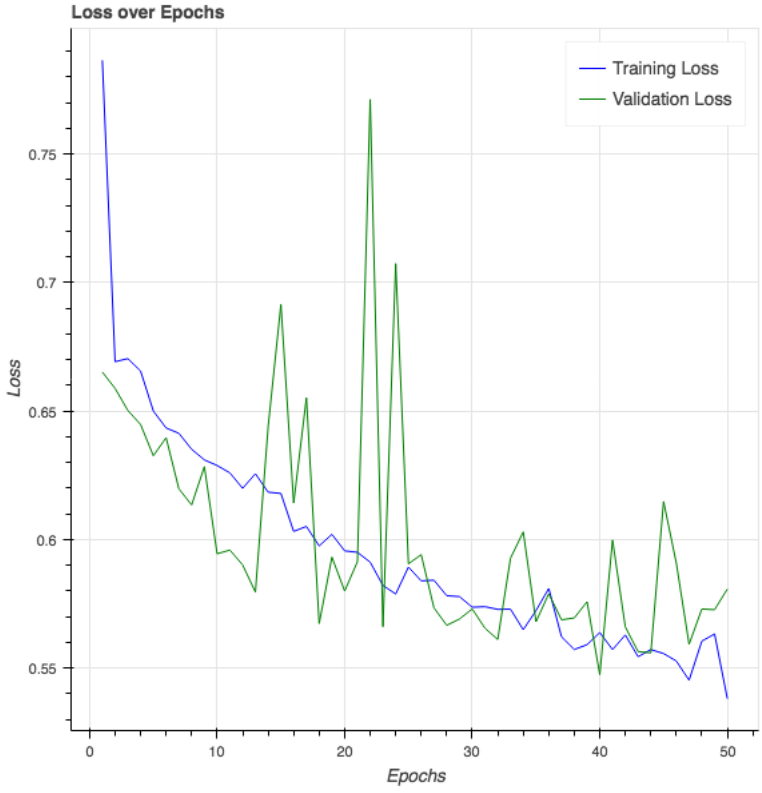

FIGURE 19. Smaller DCNN

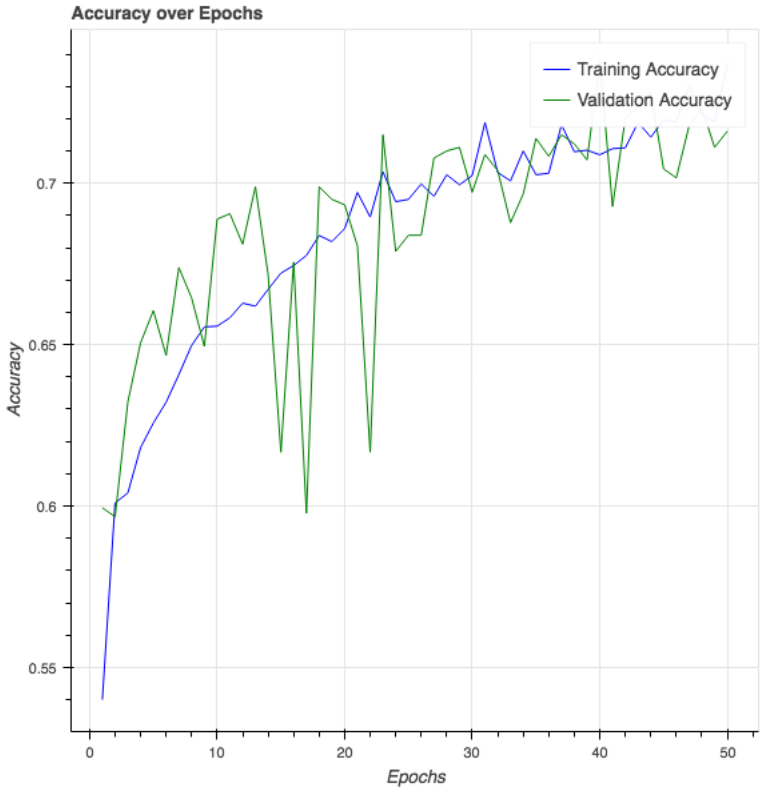

FIGURE 20. Smaller DCNN

VGG16 with Randomly Initialized Weights

In this experiment we use the VGG16 network but with randomly initialized weights. We train and validated our dataset using SGD and RMSProp. We set the optimal learning rate for SGD at 0.0001 and momentum at 0.9, whereas RMSProp has an adaptive learning rate. The augmentation parameters were the same as above. Due to our last result, The experiment with RMSProp and SGD did not converge where the training loss was reduced to only 0.55 with an accuracy of 0.75. The validation loss and accuracy was at 0.57 and 0.75 respectively as seen in figure 21 and 22

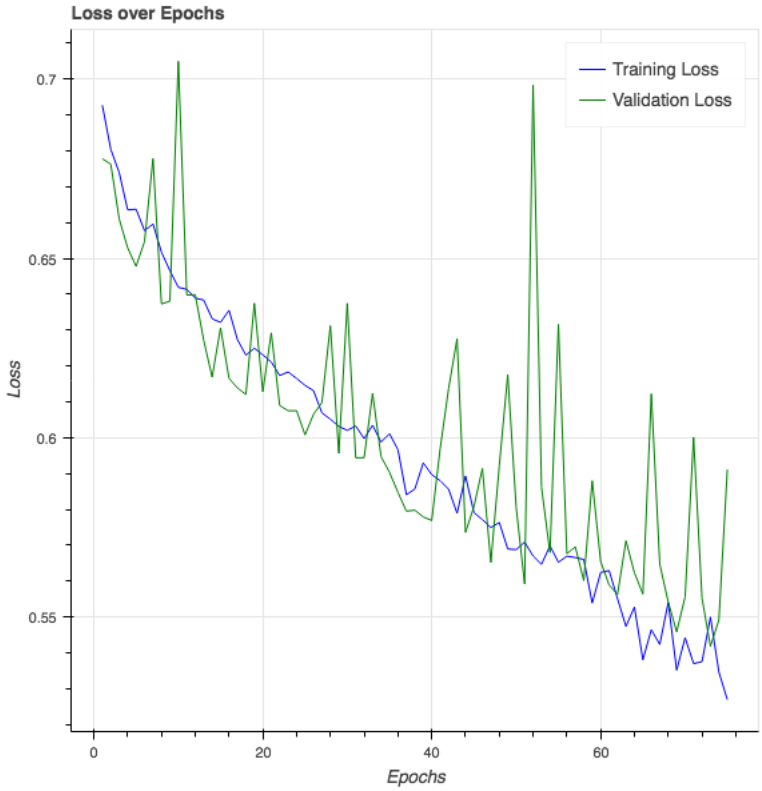

FIGURE 21. Randomly Initialized Weights, Loss - VGG16

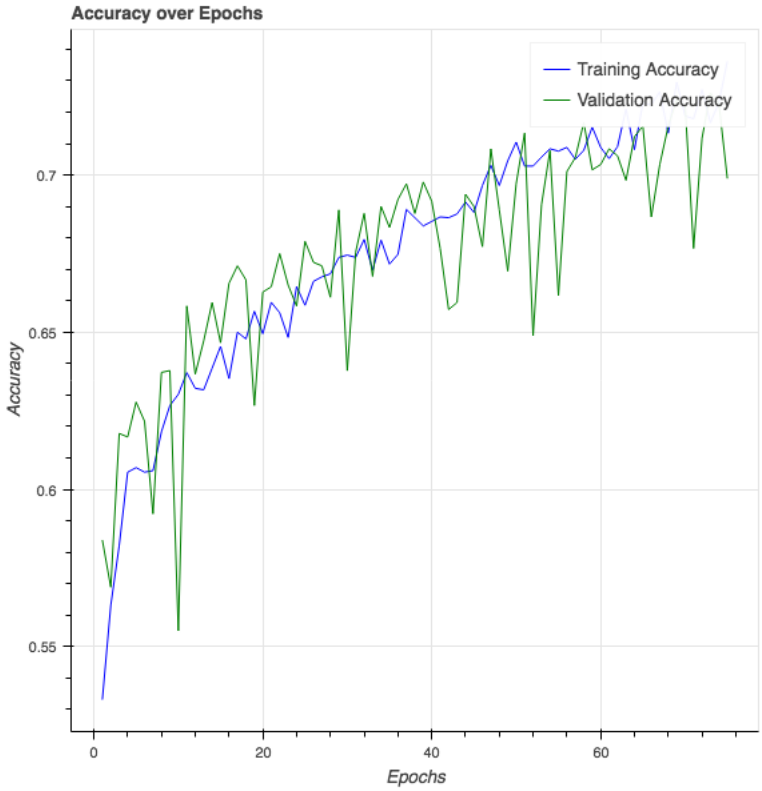

FIGURE 22. Randomly Initialized Weights, Accuracy - VGG16

Observations

Both of the above networks do not converge even when the number of parameters (layers) is increased with the VGG16 architecture, and when the number of epochs is increased. This proves that our dataset is indeed limited such that these networks of different parameters are unable to learn the required features.

VGG16: Frozen Method

With the frozen method, the final classifier is unable to learn the features extracted from the convolutional layers pre-trained on ImageNet.

VGG16: Fine - Tune Method

For this experiment, we use the SGD backpropagation algorithm along with binary cross-entropy as our loss function using a sigmoid activation in the final layer. The network was trained and validated over 75 epochs. We found that we were able to achieve the best possible accuracy after freezing only all the layers. As we tuned the layer to be frozen from the lower layers to the higher layers we saw that our validation loss reduced considerably from 0.90 down to 0.36 for an accuracy of 0.84%. When all the layers but the final layer was frozen the training loss was reduced to 0.14 and the training accuracy was at 0.95% as seen in figures 23 and 24

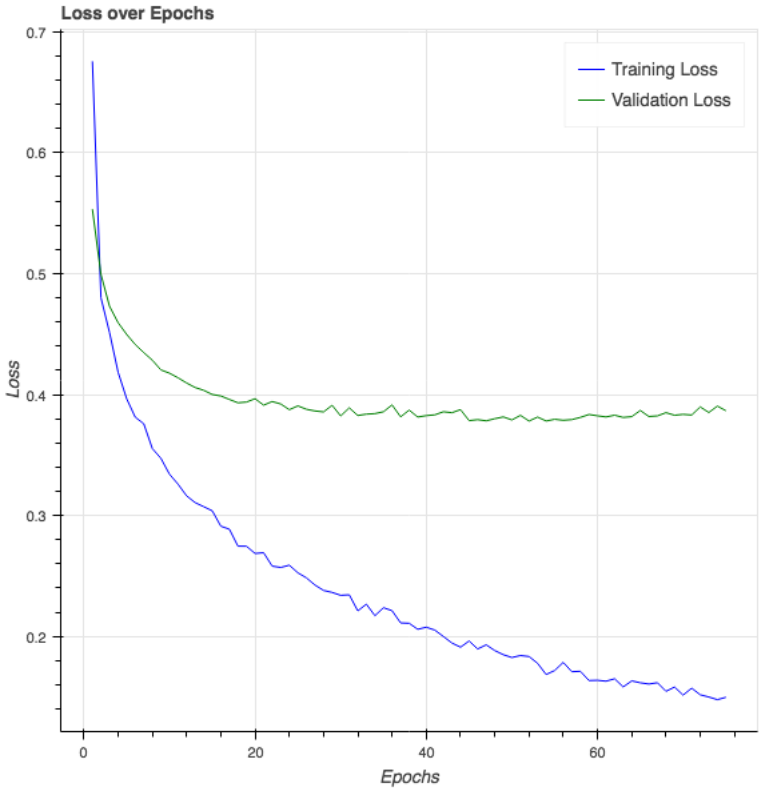

FIGURE 23. Fine Tuned, Loss - VGG16

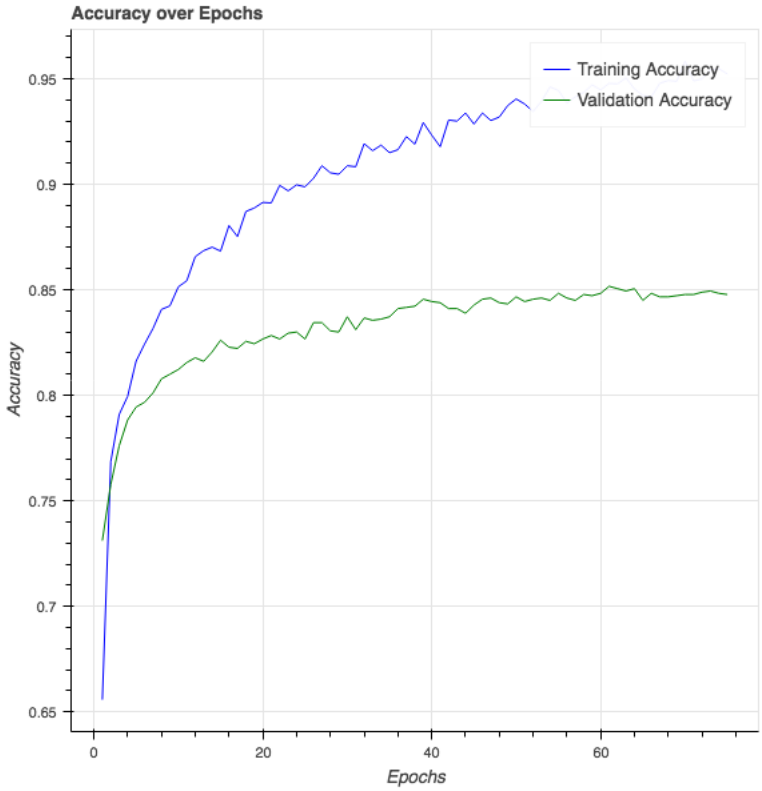

FIGURE 24. Fine Tuned, Accuracy - VGG16

The figure 25 shows the graph of fine-tuning the VGG16 network.

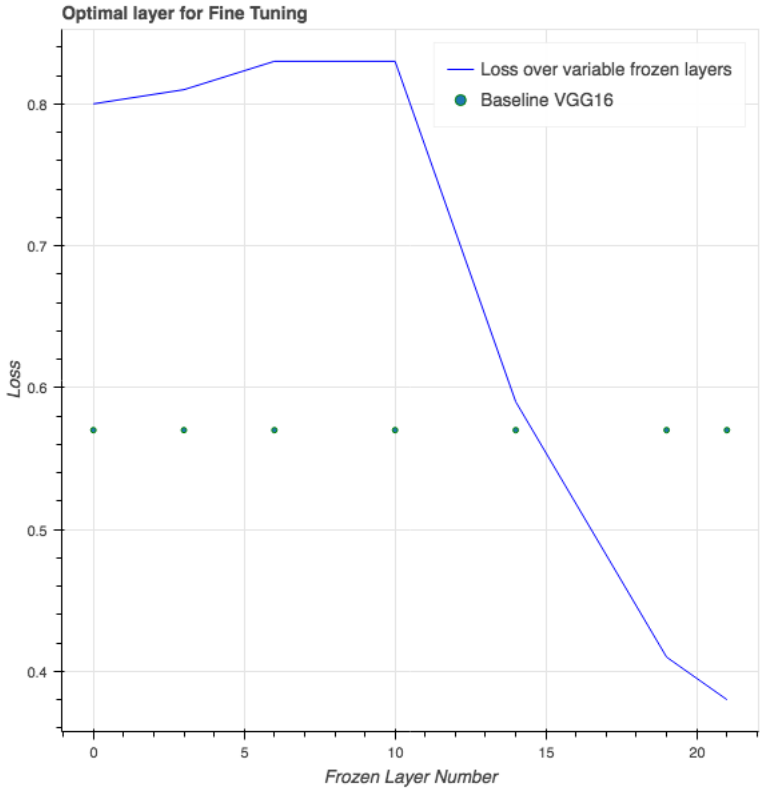

FIGURE 25. Optimal Layer for Fine Tuning - VGG16

Inception V3: Frozen Method

In this experiment using the frozen method, a fully connected layer is added, initialized with random weights, so that it can act as our classifier. Without adding this layer we could not train the network using the frozen method. The classifier is trained using RMSProp with just our dataset so that it learns the weights relative to our dataset which is faster than SGD since it uses an adaptive learning rate. It is trained to only 20 epochs since we wanted our final classifier to have weights pre-trained on our dataset which would be better than randomly initialized weights. We then use our convolutional layers as feature extractors which are given as input to the classifier. Similar to the VGG16 frozen method experiment our final classifier was unable to learn the features of our dataset extracted from ImageNet features.

Inception V3: Fine-Tune Method

In this experiment, we do not have the structural simplicity of VGG16 and therefore it takes a lot longer to empirically find the best layer to freeze until in the network. We observe the best performance when the network is frozen until the fourteenth layer as the training loss is down to 0.3 and accuracy up to 0.87% and the validation loss is down to 0.35 and accuracy at 0.85% as seen in figure 26 and figure 27

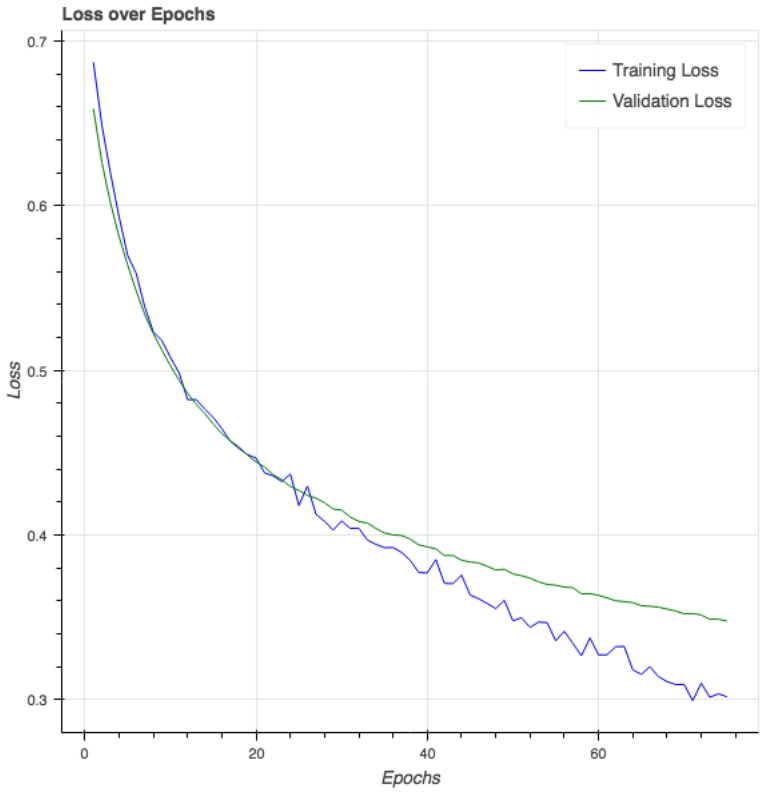

FIGURE 26. Fine Tuning, Loss - Inception V3

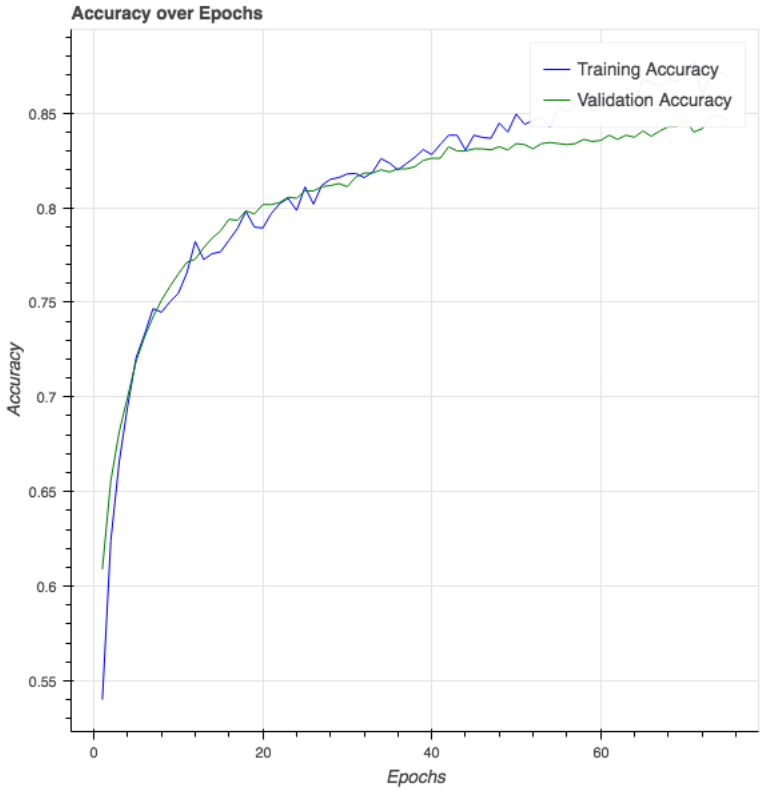

FIGURE 27. Fine Tuning, Accuracy - Inception V3

The optimal performance of fine-tuning is shown in figure 28

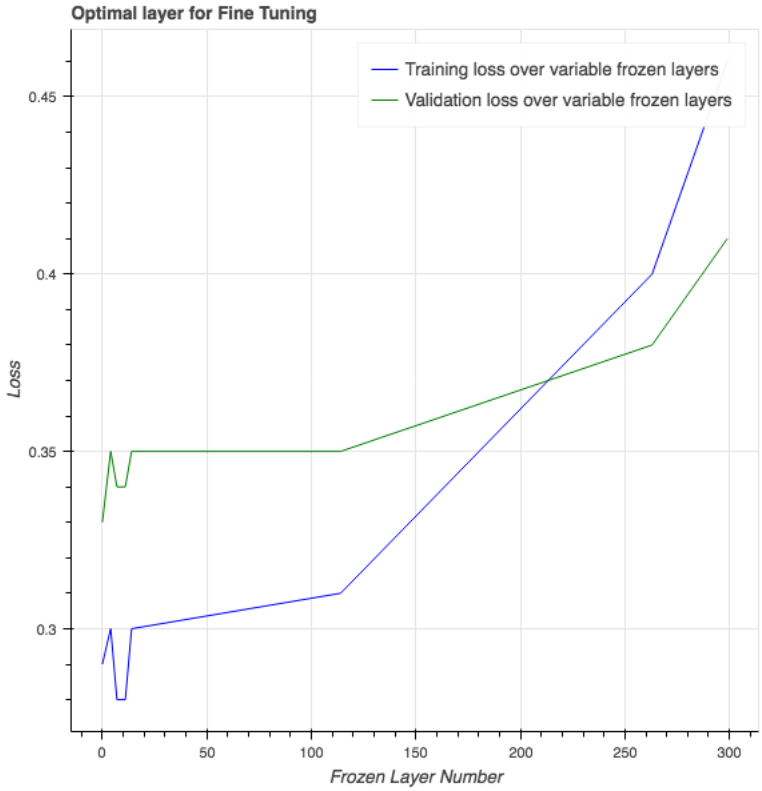

FIGURE 28. Optimal Layer for Fine Tuning - Inception V3

Observation

From the experiments in section 6.4 and section 6.4 we observe that just the ImageNet features are not enough to classify the images in our dataset irrespective of the architecture we use. Hence fine tuning is the appropriate method for our task. The previously discussed architectural simplicity helped us find the optimal layer until which its weights need to be frozen to get the lowest validation loss. It can be seen from the figure 25 that as we freeze more layers the network holds on to the features related to ImageNet, and not generalize over our dataset. Since our dataset is small the network is unable to learn the higher-order features from our dataset such that it performs the best on ImageNet features. When we compare our fine-tuning method to that of our frozen method we observe that the final classifier

of the fine-tuning method is able to learn unlike the frozen method because it is pretrained on the ImageNet dataset. The weights updated by training the final classifier with only our dataset was not sufficient for the network to learn.

From the figure 28, we can observe that we obtain the lowest validation loss from the networks that are frozen until only its first few layers. This means that unlike VGG16, the Inception V3 architecture for our task uses generic features of ImageNet only until the first few layers. The rest of the network is allowed to fine tune and learn the representation of our dataset. As we increase the layer until which the weights are frozen we observed that the validation loss reduces to lower than that of the training loss which means that the network is able to perform on the features of ImageNet but the overall loss suffers as the network is unable to learn and classify the representation of our images.

From the experiments in section 6.4 and section 6.4 it can be seen that Inception V3 performs better than the VGG16 experiments as we see considerable overfitting. Since Inception V3 was able to outperform VGG16 on our task, we can state that the depth of the Inception V3 network was required to be able to learn or fine-tune from the ImageNet features. We applied dropout to reduce the overfitting as mentioned in section 5.2 though there was only a meager difference in the validation loss, and the network performance degraded in terms of training loss.

From the experiments discussed above, we observe that the performance, especially in terms of accuracy is lower than that of the state-of-the-art OR models. It can be argued that the OR models have datasets such as ImageNet that are curated by the community and are geared towards object recognition, and such a dataset is not yet available for our task. The dataset is another reason why the accuracy is much lower since it is a weakly labeled dataset as it is not verified by

humans. It contains noise in terms of the class labels of the images which may not be absolute and may be disputed. Though we are able to show that our DCNN was still able to learn up to a certain extent the differences between dynamism and its absence with respect to the distribution of images in our dataset. The performance of these DCNNs would only improve given a strongly labeled dataset of images.

Effect of Dynamism on Emotion

As our Inception V3 implementation gives us the best results we use the network as a classifier to classify the images as dynamic or 'still' in the IAPS and OASIS datasets.

Effect on IAPS Dataset

In figure 29 is the plot of the number of images classified as dynamic on the valence ratings and in figure 30 is the plot of the number of images classified as 'still' on the valence ratings.

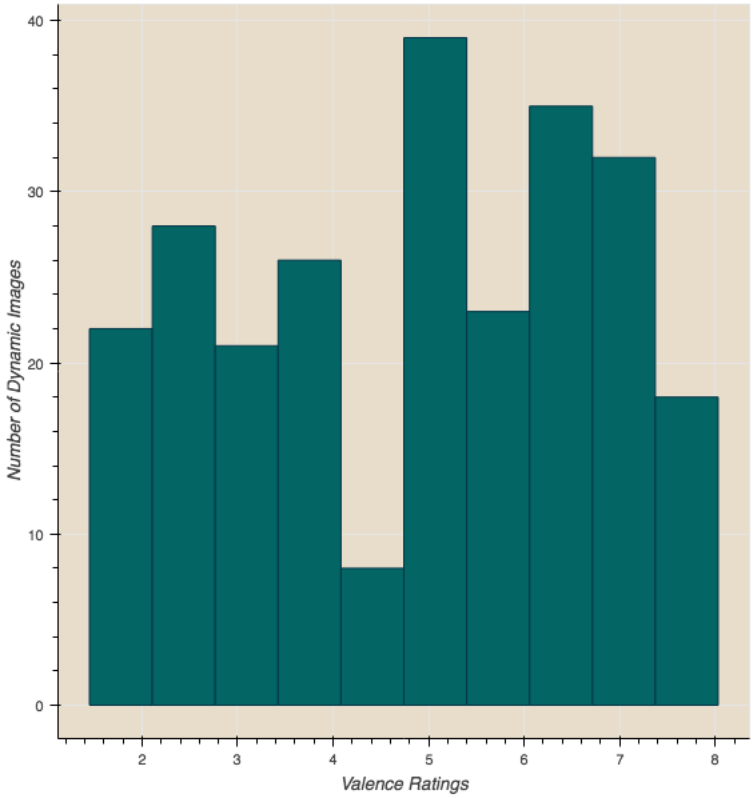

FIGURE 29. Valence Ratings of Dynamic Images - IAPS

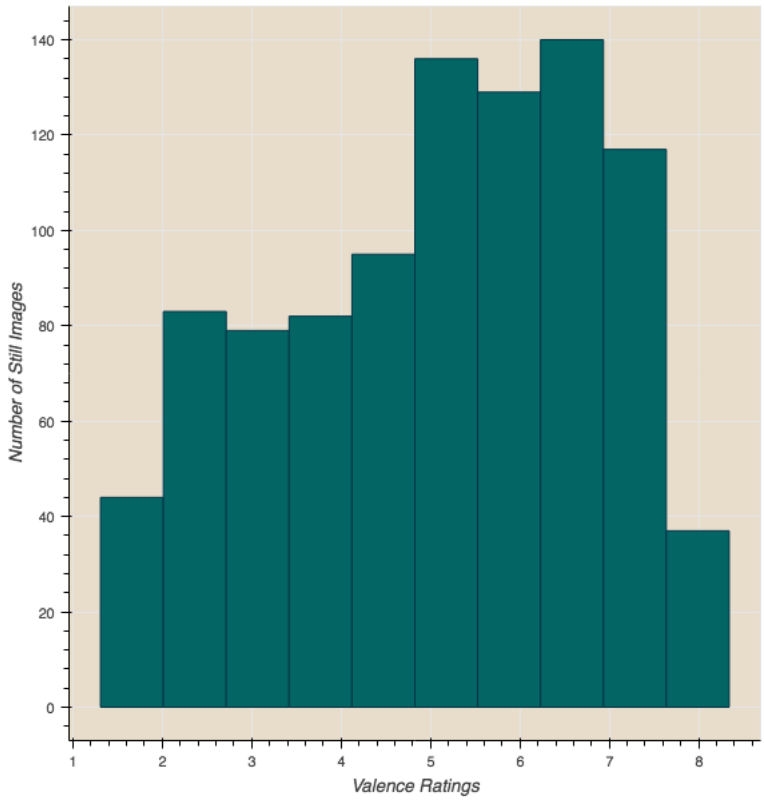

FIGURE 30. Valence Ratings of 'Still' Images - IAPS

In the figure, 31 is the plot of the number of images classified as dynamic on the arousal ratings and in figure 32 is the plot of the number of images classified as 'still' on the arousal ratings.

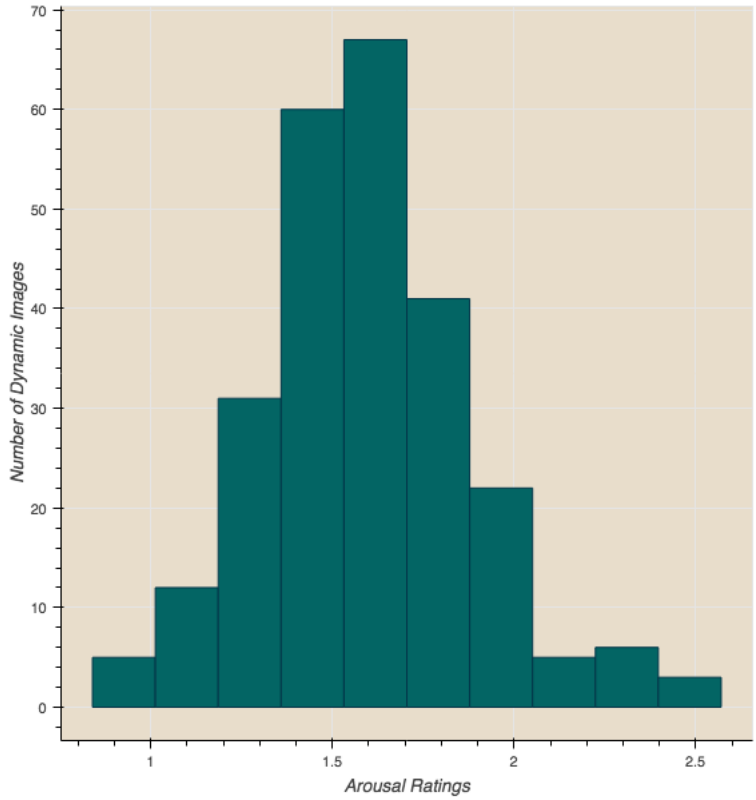

FIGURE 31. Arousal Ratings of Dynamic Images - IAPS

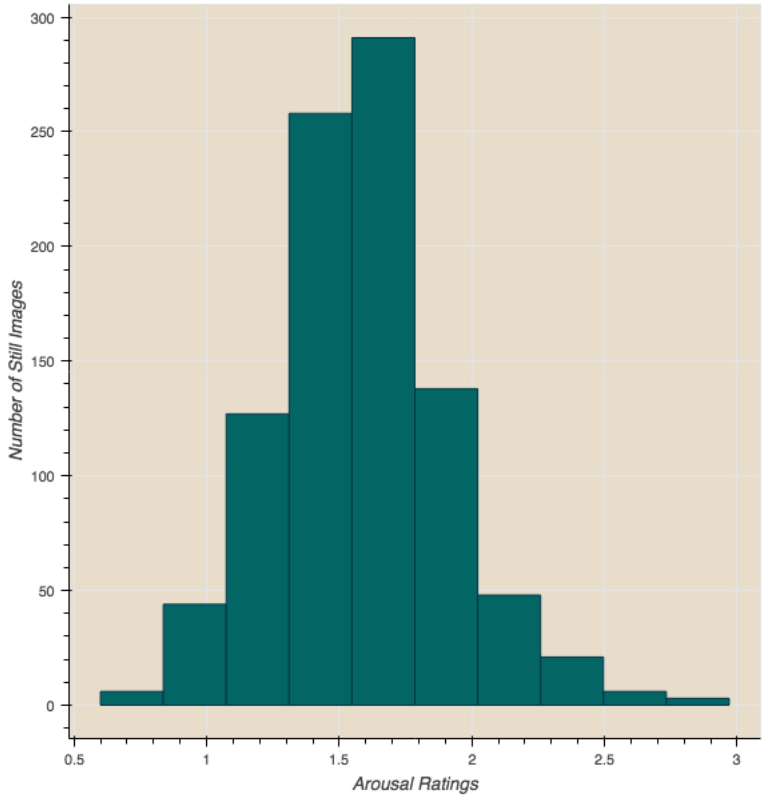

FIGURE 32. Arousal Ratings of 'Still' Images - IAPS

Effect on OASIS Dataset

In the figure, 33 is the plot of the number of images classified as dynamic on the valence ratings and in figure 34 is the plot of the number of images classified as 'still' on the valence ratings.

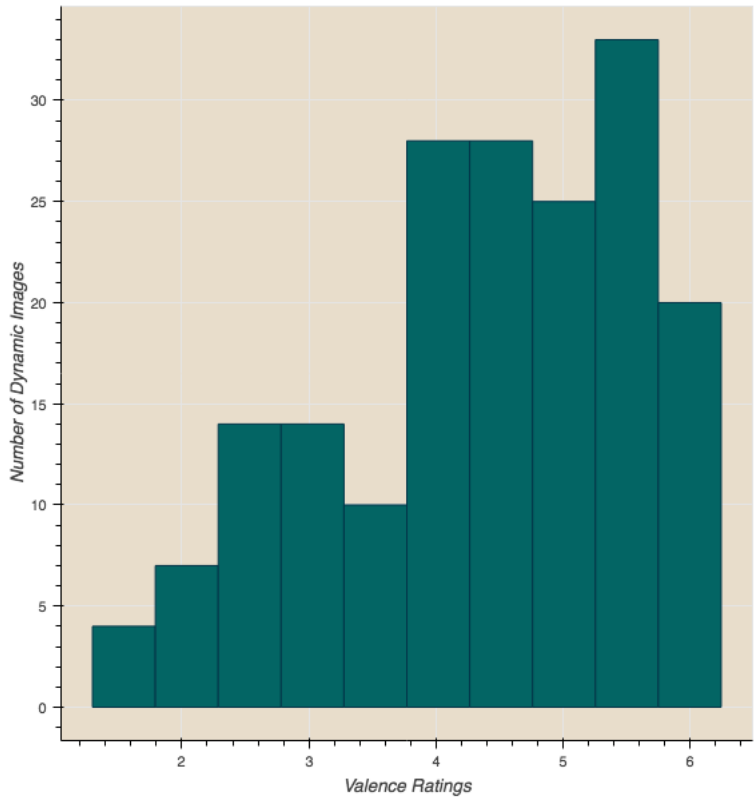

FIGURE 33. Valence Ratings of Dynamic Images - OASIS

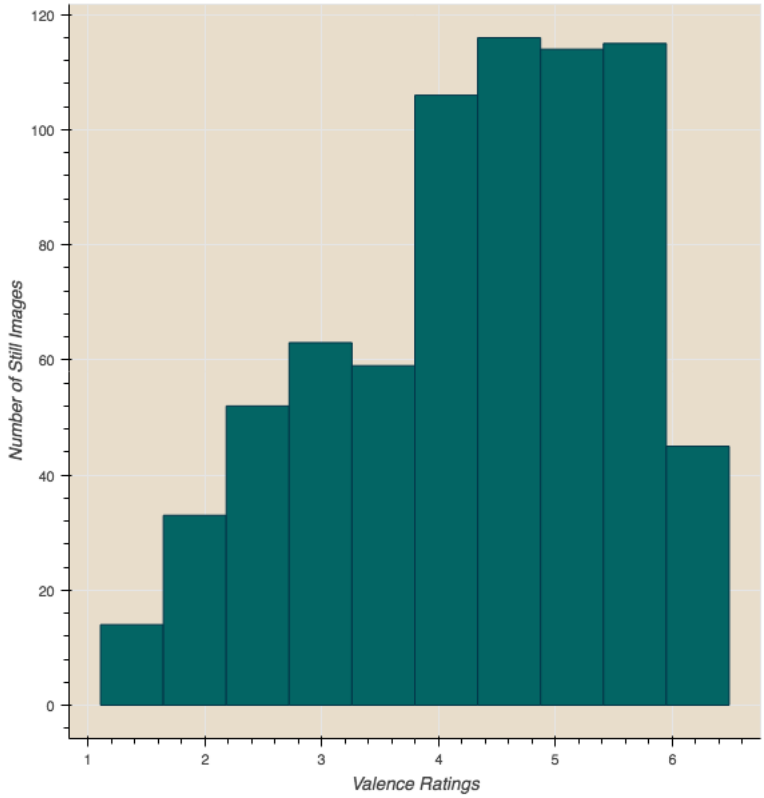

FIGURE 34. Valence Ratings of 'Still' Images - OASIS

In the figure, 35 is the plot of the number of images classified as dynamic on the arousal ratings and in figure 36 is the plot of the number of images classified as 'still' on the arousal ratings.

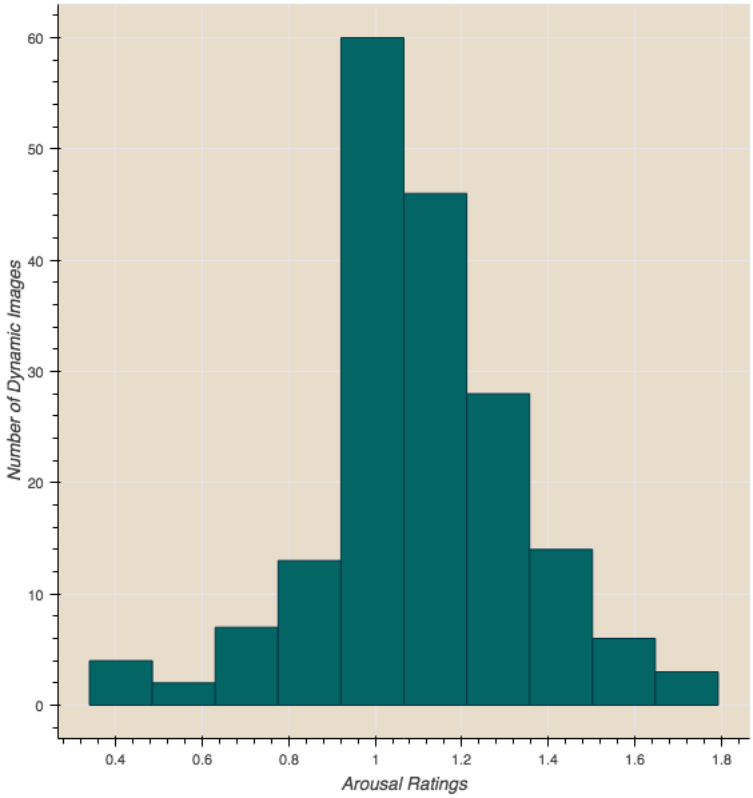

FIGURE 35. Arousal Ratings of Dynamic Images - OASIS

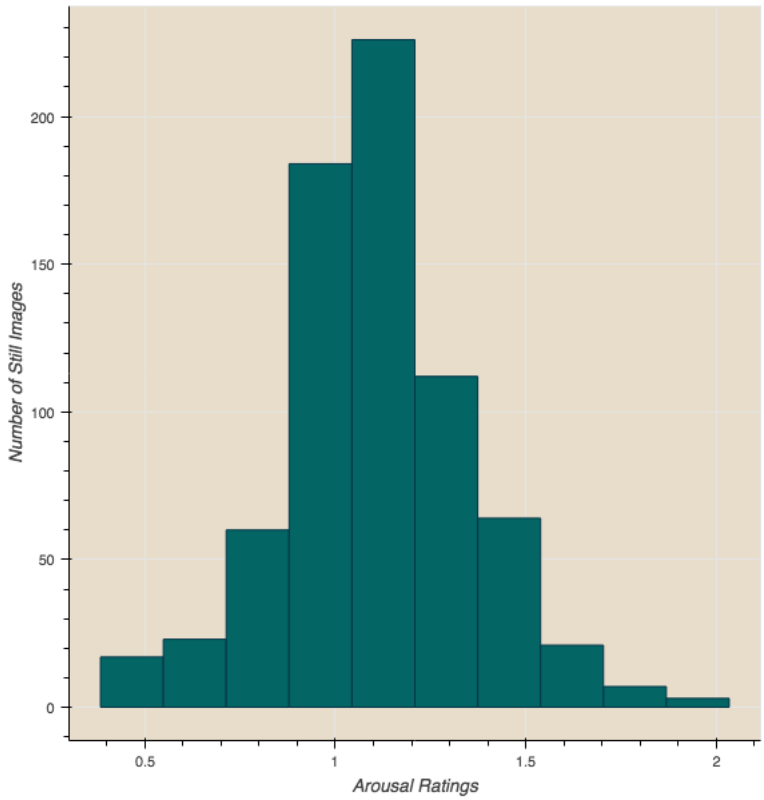

FIGURE 36. Arousal Ratings of 'Still' Images - OASIS

Observations

We used our Inception V3 DCNN to predict the dynamism or 'still' labels on the IAPS dataset. The images in this set are different from the ones that were used to train the DCNN. We found that the IAPS dataset had 20% of its images classified as dynamic and the rest as 'still'.

It can be seen from the figure 31 that the images on the arousal ratings, range from 1.55 to 1.88 have the most dynamism, and from the figure 29, that the images on the valence ratings range from 4.8 to 5.4. This means that dynamic images are found more on the neutral side of the mean intensities of the images in IAPS and are lesser on the extreme ratings of arousal. On the valence ratings, dynamic

images are more fairly distributed with higher frequencies from 4.8 to 5.4 of the valence ratings.

For 'still' images in figure 32, they have a higher frequency on the same range of arousal ratings as images with dynamism. This seems counter-intuitive as 'still' images and dynamic images are on a similar range. 'Still' images are also much lesser in frequency on the extreme ratings of arousal. Though, we see a higher frequency of 'still' images than dynamic images on the same range of arousal ratings. For 'still' images, as seen in figure 30 higher frequencies are seen on the range of 5 to 7 on the valence ratings. The dynamic images on the valence ratings are not as well distributed as the 'still' images on the valence ratings. A drop in the number of dynamic images can be seen in figure 30 around the rating 5 on the valence ratings.

We use the same DCNN to predict the labels of the OASIS dataset. Similar to the IAPS dataset, we found that 20% of the images were labeled as dynamic.

We can see from figure 35 that the dynamic images have a higher frequency in the range of 0.9 to 1.1 on the arousal scale, and range of 5.2 to 5.8 on the valence scale, as seen in figure 33. Similar to the distribution of images in the IAPS set we see in figure 35 that dynamic images are in higher frequencies at the neutral side of the mean arousal ratings and are lesser on the extreme scales of the arousal ratings. On the valence ratings, dynamic images were found to be in higher frequencies towards the higher valence ratings indicating that dynamic images evoke a higher level of pleasure.

From the figure 36, it can be seen that the 'still' images have a higher frequency from 1.1 to 1.2 ratings on the arousal scale. When compared to dynamic and 'still' images on the arousal ratings, we see the same statistics as the IAPS

dataset. The 'still' images also have lower frequencies on the extreme sides of the arousal ratings. It is observed that for the same arousal ratings, 'still' images have a higher frequency compared to dynamic images. From the figure 34 it can be seen that images are found to be 'still' at a higher valence rating of 4.5 to 6 which is counterintuitive to the dynamic images found on the valence ratings in figure 33

From the above observations, it can be concluded that there are counterintuitive results regarding the higher frequencies of dynamic and 'still' images in both the datasets. We attribute these results to our DCNN which has been trained on a different distribution of images compared to the IAPS and OASIS datasets. We expect to find a better distinction in the ratings of dynamic and 'still' images when the test images are drawn from the same distribution as the images used for training the DCNN.

CHAPTER VII

CONCLUSION AND FUTURE WORK

Conclusion

To conclude, we have studied and understood the definition of perception of the sense of movement in static visuals like images, its implications in the real world, especially towards engagement and attitude. We have explained methods to find a solution to the said problem by looking at another similar problem in a domain of AFIC. Understanding the methods from AFIC we were able to construct a pipeline of methods to solve the said problem. We follow that with a study of the theory and empirical study to understand the concept of transferring knowledge from one domain to another and employ methods that best fit our problem. Towards understanding the methods that we employ, we delve into the study of the network architectures we use and the spatial invariance we introduce using data augmentation to avoid over-fitting. We gather our own dataset and collect two other datasets to understand the effect of perception of the sense of movement or dynamism on emotions. Then we conducted different experiments to find the optimal way to train a classifier that has learned the ability to differentiate between dynamism and still static visuals as mentioned in the introduction. We then gauge our classifier to understand dynamism on the emotional scale of valence and arousal.

Future Work

Given that we have a classifier that is able to differentiate between dynamism and its absence there are a couple of ways we are already looking to improve the performance of the said classifier. We have identified certain methods of improvements that we think could improve the model and find a correlation between dynamism and emotions like we tried to do with this one.

Given that we now have a classifier that has the ability to differentiate between the perceived sense of depth and its absence up to a certain extent. There are a couple of methods we have identified to improve the performance of our classifier. These improvements would also ultimately lead to a better understanding of dynamism on the emotional scale of valence and arousal.

We look forward to improving the quality of the dataset by obtaining a dataset that is manually verified by using human intelligence. Since we use supervised learning, this would certainly improve the performance of our DCNNs. Additionally, we intend to reduce the category variety of the images in the dataset such that the model is able to learn the difference of sense of movement for the same given object. Such datasets can also be created for different category types like objects, humans, animals etc and for datasets with different demographics like product advertisements. Our current work deals with building a classifier to understand only the difference between a sense of movement and absence. We further intend to study the different types of movement and its degree within static visuals.

There are other DCNN architectures that employ different design decisions which could improve the scale of performance. They have different depths of layers, hyperparameters, and optimizations. Due to the very black box nature of DCNNs,

the performance differential can only be determined empirically for our tasks. These models can also be adapted for resource-constrained environments such that their availability and usage is not constrained by available resources. Towards the study of understanding the effect of dynamism on emotions, we intend to construct handcrafted features from domain experts such that dynamism or its degree could be altered to affect emotion and other factors like engagement, attitude etc.

An investment into such datasets, architectures and studies would spur research within the domain of our task but even other tasks that deal with learning the stimuli that affect a viewer of static visuals. A collaboration with researchers from consumer psychology would help us build extensive methods to test our models and build applications that have real-world effects. It would then be possible to compare the performance of our models to those that employ human intelligence.

REFERENCES CITED

Borth, D., Ji, R., Chen, T., Breuel, T., and Chang, S.-F. (2013). Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 223–232. ACM.

Bradley, M. M. and Lang, P. J. (1994). Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Cian, L., Krishna, A., and Elder, R. S. (2014). This logo moves me: Dynamic imagery from static images. *Journal of Marketing Research*, 51(2):184–197.

Craig Lefebvre, R., Tada, Y., Hilfiker, S. W., and Baur, C. (2010). The assessment of user engagement with ehealth content: The ehealth engagement scale. *Journal of Computer-Mediated Communication*, 15(4):666–681.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009a). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009b). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Dondis, D. A. (1974). *A primer of visual literacy*. Mit Press.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, (2):303–338.

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Howard, A. G. (2013). Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM.

Kim, H.-R., Kim, Y.-S., Kim, S. J., and Lee, I.-K. (2018). Building emotional machines: Recognizing image emotions through deep neural networks. *IEEE Transactions on Multimedia*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

Kurdi, B., Lozano, S., and Banaji, M. R. (2017). Introducing the open affective standardized image set (oasis). *Behavior research methods*, 49(2):457–470.

Lang, P.J., B. M. . C. B. (2008). International affective picture system (IAPS): Affective ratings of pictures and instruction manual. *Technical Report A-8*.

Leborg, C. (2006). *Visual grammar*. Princeton Architectural Press.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Machajdik, J. and Hanbury, A. (2010). Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 83–92. ACM.

Mikels, J. A., Fredrickson, B. L., Larkin, G. R., Lindberg, C. M., Maglio, S. J., and Reuter-Lorenz, P. A. (2005). Emotional category data on images from the international affective picture system. *Behavior research methods*, 37(4):626–630.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1717–1724. IEEE.

Osgood, C. E. (1952). The nature and measurement of meaning. *Psychological bulletin*, 49(3):197.

Pavan, A., Cuturi, L. F., Maniglia, M., Casco, C., and Campana, G. (2011). Implied motion from static photographs influences the perceived position of stationary objects. *Vision research*, 51(1):187–194.

Pieters, R. and Wedel, M. (2007). Goal control of attention to advertising: The yarbus implication. *Journal of consumer research*, 34(2):224–233.

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

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. corr abs/1409.4842 (2014).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Targ, S., Almeida, D., and Lyman, K. (2016). Resnet in resnet: generalizing residual architectures. *arXiv preprint arXiv:1603.08029*.

Teixeira, T., Wedel, M., and Pieters, R. (2012). Emotion-induced engagement in internet video advertisements. *Journal of Marketing Research*, 49(2):144–159.

Valdez, P. and Mehrabian, A. (1994). Effects of color on emotions. *Journal of experimental psychology: General*, 123(4):394.

Warriner, A. B., Kuperman, V., and Brysbaert, M. (2013). Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Wu, Z., Shen, C., and Hengel, A. v. d. (2016). Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

You, Q., Luo, J., Jin, H., and Yang, J. (2016). Building a large scale dataset for image emotion recognition: The fine print and the benchmark. In *AAAI*, pages 308–314.

Yu, F. X., Cao, L., Feris, R. S., Smith, J. R., and Chang, S.-F. (2013). Designing category-level attributes for discriminative visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 771–778.