Interpreting Neural Networks via Activation Maximization

Vitaly Volozhinov

Supervisor(s): Dr. Adrian Muscat

Faculty of ICT

University of Malta

May 2019

Submitted in partial fulfillment of the requirements for the degree of B.Sc. Computing Science (Hons.)

Abstract:

Decision trees are models whose structure allows for tracing an explanation of how the final decision was taken. Neural networks known as 'black box' model, do not readily and explicitly offer an explanation of how the decision was reached. However since Neural Networks are capable of learning knowledge representation it will be very useful to develop methods that interpret the model's decisions.

In this project Activation Maximisation will be used to search for prototypical inputs that maximise the model's response for a quantity of interest. A pair-wise prototype comparison is then carried out under different learning conditions, such as number of classes the model deals with. The study is grounded in the area of object spatial relations recognition in images and will shed light on what models are learning about objects in 2D images which should give insight into how the system can be improved.

The spatial relation problem is one where given a subject and an object the correct spatial preposition is predicted. This problem extends beyond just predicting one correct spatial preposition as there are mulitple possible relationships associated between two objects.

Contents

1 Introduction

Research in computer vision has excelled in recent years largely due to technological advancements in hardware. This allowed for more computationally intensive ideas to be explored and implemented. Deep Learning a subset of Machine Learning is one of those ideas based on artificial neural networks. Deep learning in computer vision comes in the form of Convolutional Neural Networks (CNN's). This form of Machine Learning is effective as it takes the given images and through a sequence of convolutions and pooling layers transforms this data into that of smaller size while retaining important features. This reduces the training time and computational power required to classify images compared to a Neural Network.

CNNs have become very precise and effective in solving the problems of object detection and localization in images. Up until recently it was thought to be impossible for a computer to distinguish between a cat and a dog in an image due to them having similar general features but nowadays anyone can implement a simple CNN to solve this classification problem. The focus is shifting from object recognition to the study of visual relationships between objects in an image. This is the visual relationship detection (VRD) problem. In this problem given a subject and an object, the machine learning model must predict the best predicate that describes the visual relationship between those two objects.

The VRD problem was first tackled by Sadeghi and Farhadis (2011) by taking triplet representation $< subject, predicate, object >$ as a class, this has led to an exponential growth in classes and a long tail distribution problem. A solution to that was to divide the problem up into parts. The first part would be to perform object detection on the two objects and then pass their Union into a new network which was specialized in predicate prediction as done by Lu et al (2016). There have been improvements to accuracy for this method such as having geometric and text features accompany the CNN model for increased accuracy. VRD is important as it would give greater context to images which would provide real world solutions to problems such as giving audio descriptions to blind people of the environment around them.

Since neural networks are a black box models to know if a CNN is working correctly it is evaluated over unseen data and its accuracy is measured over how well it predicts this data. This is a working method however we don't understand how and why the final descion was made. It would be reassuring to understand why the decision was made. An example of why this is important, in the news there was a lady who fell asleep at the wheel of a self driving car and this car didnt stop when a pedestrian was crossing the road and

hit them. When the company looked at the logs of the car to figure out what went wrong and why the car didnt see the pedestrian they found out that the car had seen the person and decided not to stop. If all the descion making process of the car had been carefully understood and analyzed the people creating the A.I could have seen that in one of those decision paths the car would see the person and not stop. As A.I systems are being integrated into our daily lives such as medical diagnosis and driverless cars, it is important to make sure they are understood and Activation Maximization is one of the methods that will be explored in this dissertation.

The research aim is to use Activation Maximization on the VRD problem to interpret and understand what the CNN is looking for when classifiying relations. Since VRD contains many relationships the main focus will be on spatial relations between two objects. This dissertation will focus on interpreting different models and configurations to have an understanding of what the model is learning and whether or not Activation Maximization is a useful method of doing so.

2 Background and Literature Review

2.1 Preamable

In this chapter we will be going through the components that make up a CNN and the different architectures of CNNs. This chapter also includes literature reviews on existing works tackling the VRD problem and Activation Maximization.

2.2 Convolutional Neural Networks

2.2.1 Image Components

RGB images consists of 3 channels Red, Green and Blue. These 3 channels are each represtented by a 2D array with each entry being a pixel value that ranges from 0 to 255. The pixel value in the array represents the colour intensity of the channel. Combining these 3 arrays together will yield the original image. The image details are represented as (Height ,Width,Channels) and this is the shape that the CNN expects as input. This input shape is kept the same for all images during training.

2.2.2 Convolutional Layer

The convolutional layer is composed of a kernel/filter which is a 2D matrix of a given size e.g (3x3) that performs matrix multiplication between itself and a portion of a region of the image. It does this by striding from left to right with a certain stride range e.g (Stride = 1) until the entire image is traversed. This extracts features from the image, for the first convolutional layer it would extract low level features such as colours and edges as the CNN gets deeper with more layers the features will increase in complexity to become high level features such as the wheels of car, ears of a cat, tail of a dog etc. The kernel performs two types of operations on the image, one where the feature dimensionality is reduced compared to input and another where it is increased or stays the same due to padding. Padding is when the image width and height is increased and the pixel values that have been added are filled with 0, e.g Padding = 1 would increase the image width and height by 2. This is done as there could be valuable information in the pixel values on the outskirts of the image which would otherwise be lost as the kernel would combine them with the inner pixels.

2.2.3 ReLU Layer

After a convolutional layer there is an activation function, normally this function is the ReLU (rectified linear unit). This applies the activation function f(x)=max(0,x) which removes negative values by setting them to 0. The effect of this is that the CNN learns much faster as it increases the nonlinear properties of the decision function by allowing negative value through(as they are set to 0).

2.2.4 Pooling Layer

The pooling layer is used to decrease the computational power required to process data by reducing the spatial size of the convoluted features. The dominant features are extracted without losing major information and keeping the training model effective. The two types of pooling are average pooling which returns the average of all the values from the region and max pooling which returns the maximum value. Max pooling performs better than average pooling as average pooling mostly reduces the dimensionality but max pooling acts as a noise suppressant by discarding low values (Noise).

2.2.5 Fully Connected Layer

After a multitude of Convolutional,ReLU and Pooling layers the final output is passed through a fully connected layer which is where the high-level reasoning and decision making is done. Here the output from the last pooling layer is Flattened meaning the image output goes from a 2D array to a 1D array by concatenating the rows from underneath each other to the right side of the rows above. Then all the values are passed into neurons in a fully connected layer which all have connections to the activations as done in a regular Neural Network.

2.2.6 Final Layer

The last layer is the output layer where all the neurons from the fully connected layer connect to. They connect to a specifically set amount of neurons which is defined by the amount of classes the CNN is being trained for. This last Densly Connected layer has an activation function. The final activation function of the CNN depends on the problem being solved. In this dissertation we will be focusing on two types of problems a Single Label Classification (SLC) problem and a Multi Label Classification (MLC) problem.

2.2.7 Dropout Layer

This layer is used to reduce overfitting of the training data on the model being trained. Overfitting is when the model doesnt generalize the parameters enough and represents the training data too much, meaning it would keep getting increasingly better accuracy during training but would have decreasing accuracy on the validation data. The dropout layer combats this by disabling neurons by setting them to 0 at each training stage which have a probability less than $p = 0.5$. This increases generalization as it forces the layer to learn the same concept with new neurons.

2.2.8 Single vs Multi Label Classification

The SLC problem consists of having only one correct label associated to the input. This problem would require for the last activation function to be a Softmax function. Softmax outputs a range of probabilities per class that all add up to 100%, the highest probability is considered as the best label.
The MLC problem consists of having multiple correct labels associated to the input. This problem would require for the last activation function to be a Sigmoid function. Sigmoid

outputs a range of probabilities per class where each class ranges from 0% to 100% individually and independant of the other classes. Unlike the Softmax output the probabilities do not necessarily add up to 100% therefore a threshold is usually set to what is an accepted output or not e.g anything above 50% probability is accepted.

2.2.9 Training and Terminology

Once the structure of the CNN is setup it needs parameters for it to be compiled. The parameters consist of a loss function,optimizer and a metric. The loss function dictates how the model is penalized when the predicted values deviate from the true values. The optimizers job is to make sure that the loss function is minimized as much as possible. Finally the metric used is what shows the final accuracy of the current training cycle between the predicted and the true values. The dataset would be seperated into 3 parts training/testing/validation , the training data is used to train the CNN and the test data is used to evaluate the trained model on unseen data. The validation data is used during training as unseen data to show if the model is being overfitted during training, it can be seen that it is overfitting when the training accuracy is going up and loss is going down but the opposite is occurring for the validation accuracy and loss.

2.2.10 Very Deep Convolutional Networks For Large-Scale Image Recognition

In attempt to improve CNNs at the time, mainly the original architecture of Krizhevsky et al. (2012). The Visual Geometry Group focused on the depth of the convolutional network an important aspect which affects how high the level of the features could be learned. The architecture had its parameters fixed and the convolutional layers increased thus increasing the depth of the network, this was feasible as small kernels (3x3) had been utilized in all the layers. This method produced better results than previous architectures at the time. The two best performing models weights have been released to the public. The CNN was trained using fixed-size 224x224 RGB images which had been preprocessed by subtracting the mean RGB value from each pixel. The images had been then passed through a stack of convolutional and max pooling layers with a window of 2x2 pixels of stride 2. The CNN has 5 max pooling layers and not all the convolutional layers are followed by a max pooling layer. The convolutional strides are performed by a kernel of size (3x3) with Stride = 1 and a padding of 'Same' to preserve the spatial resolution after the convolution. Each hidden layer is equipped with a ReLU function, the architecture of VGG16 is dhown in Figure 1. After all the convolutional and max pooling layers there are

3 fully connected layers with the first two having 4096 neurons each and the last densely connected layer having 1000 output neurons for every class. The models had been trained as a SLC problem therefore the last activation function is Softmax. The top performing models had 16 and 19 layers therefore their acquired names are Visual Geometry Group 16 (VGG16) and Visual Geometry Group 19 (VGG19).

The model was trained using mini-batch processing with a batch size of 256. The loss function used was stochastic gradient descent(SGD) with a momentum of 0.9, drop out ratio of 0.5 and was trained for 74 epochs. The initial learning rate was 0.1 and was decreased by a factor of 10 when accuracy stagnated. It was trained using four NVIDIA Titan Black GPUs for two to three weeks.

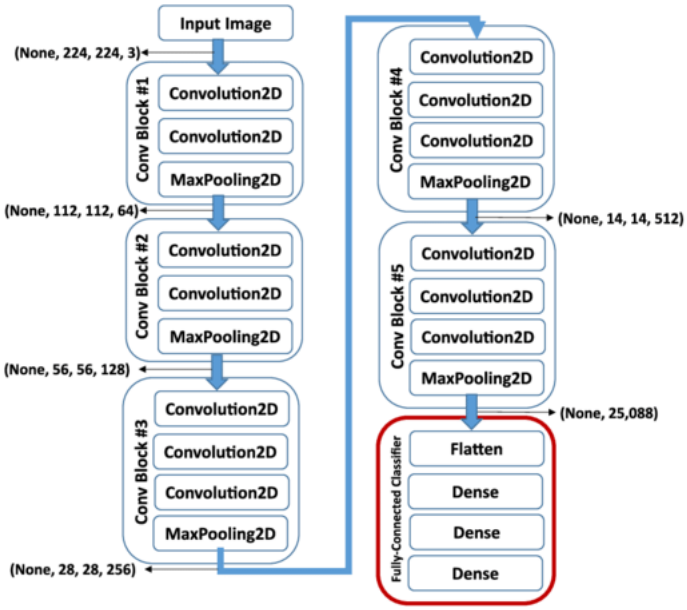

Figure 1: Architecture of VGG16

2.3 Visual Relationship Detection

Visual relationships describe interactions between objects in images. Object detection models would equal images with objects of the same type but wouldn't understand the context behind them, for example you could have two images with a dog sitting near a cat or a dog chasing a cat but for the model they would mean the same thing. The problem of classifying those relationships is the large amount of possible relationships that the same pair of objects could have between them and which best fits the description.

2.3.1 Recognition Using Visual Phrases

Sadeghi and Farhadi (2011) approached the VRD problem by taking the visual phrase $< subject, relationship, object >$ as one class. It was believed that detecting visual phrases as a whole was much easier than detecting participating objects due to the fact the objects change when participating in relations such as in $< person, riding, horse >$ the persons leg might be obscured by the horse making it harder for the system to detect them. Since one class represents a visual phrase that makes it a SLC multi-class problem. To implement this theory the Pascal VOC2008 dataset was used to extract the 8 object classes and 17 visual phrases then Bing was used to gather images for the phrases and filtered manually to keep the relevant ones. A concern was that the number of phrases grew exponentially and there wouldnt be enough training data for each visual phrase but it was thought that the number of useful visual phrases is significantly smaller than all the possible combinations. The results showed that the model achieved higher accuracies than the baseline. The problem with this is only 17 visual phrases had been used meaning that it was tested on a small dataset and it wouldnt be scalable.

2.3.2 Visual Relationship Detection with Language Priors

Lu et al 2016 showed that there is no need to have that many unique detectors by having N objects and K predicates it would take $O(N^2K)$ unique detectors as used by Sadeghi and Farhadi (2011).By separating object and relationship detection it therefore reduces the amount of unique detectors to $O(N + K)$. This solved the exponential growth of classes problem,the reason this wasn't previously implemented by Sadeghi and Farhadi (2011) is that object detection models were lacking. Another noted problem was that relationships occur in a Long Tail Distribution meaning that $< Car, On, Street >$ would be a very common occurance while $< Elephant, Drinking, Milk >$ is a rare one which makes supervised learning a problem. The proposed solution came in 2 modules the visual appearance module to solve the problem of quadratic explosion of classes and the language module to solve the long tail distribution problem. The work was done on the Visual Genome dataset which contained 33k object categories and 42k relationships categories making this a bigger dataset over the previously used one by Sadeghi and Farhadi (2011).

The visual appearance module was done by first training an object detection CNN by Fine-Tuning the VGG with (Image Net weights) to classify N = 100 object categories. Then similarly another CNN was created by Fine-Tuning a new VGG with (Image Net weights) to classify K = 70 predicates by using the Union box of two objects. The language

module was done by having relationships of similiar semantic relations be optimally mapped close together into an embedding space. The projection function (mapping) was done by firstly using word2vec (pre-trained word vectors) to have the two objects in a relationship projected into a word embedding space. Then the two vectors had been concatenated together and transformed into a relationship vector space. The relationship vector space is used to represent how all the objects interact with each other. If the language module had only seen $< person, riding, horse >$ and was shown a person riding an elephant it would predict $< person, riding, elephant >$ correctly as the word vectors of elephant and horse would be close to each other due to being rideable animals.

The process went as follows, firstly the image was passed through the object detection CNN which located objects in the image. Then a pair of those objectss bounding boxes was taken and their Union box was passed through the relationship detection CNN which predicted the probability of how likely a particular relationship is given the two bounding boxes. The probability and triplet output was then passed through the language module which then filtered out improbable relationships. The model was compared to the Visual phrases model as done by Sadeghi and Farhadi (2011) and the visual appearance model where only the visual apperances were taken into account. Due to the amount of possible combinations of objects and relationships there had been a shortage of training examples for the Visual phrase model which caused poor performance. The Visual module alone had problems discriminating against similar relationships. The full model had an 11% improvement over the visual module alone which proves that the language module from similar relationships significantly helped relationship detection.

2.3.3 Detecting Visual Relationships with Deep Relational Networks

Dai et al (2017) proposed a Deep Relational Network to statistically exploit dependencies and spatial configurations between objects and their relationships to solve the problems of having a high diversity of visual appearances for relationships and the large amount of unique visual phrases. The solution proposed by Lu et al (2016) was noted to have a problem in the visual apperance module of high diversity with different object categories sharing the same relationship predicate and even some having nothing in common. This work has contributed two main things to solve the VRD problem a DR-Net which combines stastical models with deep learning and a state-of-the-art framework for visual relationship detection.

Framework Process: Object Detection : the proposed framework works by first detect-

ing individual objects and localizing them with a bounding box and an appearance feature each. The object detector used is the Faster RCNN.

Pair filtering: For all the objects that had been detected by the Faster RCNN the next step was to produce a set of object pairs, with a total of n objects in an image there will be $n(n-1)$ possible pairs. Most of these pair combinations are meaningless therefore they are filtered out using a low-cost neural network which focuses on spatial configuration and object categories. Once the pairs are finalized they are fed into a Joint Recognition module.

Joint Recognition: A combination of the appearance module and a spatial module are used and their output is joined together in two fully connected layers. The appearance module is used on the bounding box of the image where it captures not only the object features but also its surrounding area giving more context to it. The spatial module is used by taking spatial masks from the bounding boxes and downsampling them to a size of 32x32 which are then passed into three convolutional layers to output a spatial vector.

Integrated Prediction: The compressed pair feature outputted from the fully connected layers are combined with the subject and object feature vectors and fed into the DR-Net through multiple inference units. The subject and object features are used to remove the ambiguities caused by visual or spatial cues by exploiting the statistical relations of the predicates most found between the subject and object. The DR-Net using a combination of all the data finally outputs a prediction by chosing the most probable classes for each of these components. The network was tested on the VRD and sVG datasets which produced results of 80.78% Recall@50 for VRD predicate prediction and 88.26% Recall@50 for sVG predicate prediction.

2.3.4 A Study on the Detection of Visual Relationships

This work focused on expanding Dai et als spatial masks method of preparing images for training. Before the focus had been on the Image size and the amount of Convolutional layers a CNN has but this focuses on the way the images are prepared and fed into the CNN. Several methods had been explored and tested on two different CNN architectures which were VGG16 and VGG19. The dataset used was the Stanford VRD dataset. Training was done as a SLC problem with evaluation metric of Recall@1.

Union method takes the Union of the subject and and object bounding boxes cropping the image to those dimensions. The cropped image containing the subject and object image pixels was then resized to 224 by 224 and fed into the CNNs.

Union-WB method expands on the Union method by keep the pixels within the sub-

ject and object bounding boxes but removing all the pixels that do not fall within those bounding boxes by setting them to black [0,0,0].

Union-WB-B method expands the Union-WB method by having the background set to black, subject bounding box set to green [0,255,0] and the object bounding box set to [0,0,255]. When there is an overlap of bounding boxes the pixel values are set to [0,255,255] a combination of green and blue.

Blur method the objective of this method was to exploit the lower layers of the CNN as they act as edge detectors. In this method the pixels within the subject and object bounding boxes are kept the same but the background was blurred. The Union of the two objects had been extracted and a Gaussian low-pass filter was used to blur the background. The background was blurred three ways by using standard deviations of 3(low), 5(medium) and 7(high) in the X and Y directions.

Segment method took the subject and object segmentations, created a blank image, set the pixel values of where the original objects were to that of the previously taken segmentations. This created more specific images with no noisy background.

Segment-B method expands on the segment method by setting the original subject and object pixel values to green and blue masks. This same method was used in Union-WB-B to generalize the data to only focus on visual relationships and not the objects themselves.

The VGG16 and VGG19 models had been loaded and Fine-tuned to these datasets and evaluated using the Recall@1 metric. Together with the evaluation metrics the trained CNNs had been interpreted using Activation maximization and CAMs.

The significant results showed that on both CNN types the Union-WB-B method outperformed all other methods. The reason behind this is that using Green and Blue spatial masks for the subject and objects would generalize relationships throughout different object categories. Having Union-WB-B outperform Segment-B meant that the Segment-B method was too specific and that the CNNs prefer a more general input.

2.4 Datasets

2.4.1 SpatialVOC2K: A Multilingual Dataset of Images with Annotations and Features for Spatial Relations between Objects

Muscat et al 2018 came out with a dataset SpatialVOC2k which is a multilingual dataset focused on a portion of the VRD problem mainly the spatial relations. It was adapted from the PASCAL VOC2008 dataset by extracting 2026 images. These images had been

chosen as they had 2 or more objects with given bounding boxes making this datasets main focus be a multilabel dataset. This dataset also proposed 18 Geometric features (Table 1) which proved to be useful for classification together with multiple models for training and evaluating this data. This dataset doesnt only focus on the VRD problem but also on the Depth prediction problem of objects. This dataset contains 17 English prepositions and 17 French ones, the process was done by translating the english prepositions into French and then eliminating those prepositions that had fewer than 3 examples. Figure 2 shows the occurance distribution of classes in the dataset.

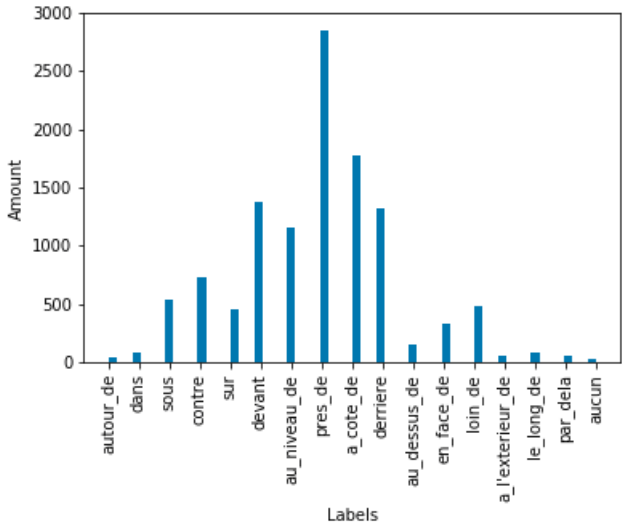

Figure 2: Class distribution SpatialVoc2k

2.4.2 Stanford VRD

The Stanford VRD dataset was introduced by Lu et al(2016). This dataset contains 5000 images with 100 object categories and 70 predicates. In total this dataset has 78872 single labels, 4504 examples with 2 labels, 685 examples with 3 labels and 71 examples with 4 or more labels. This distribution makes it mostly a SLC problem. Figure 3 shows the occurance distribution of classes in the dataset.

2.5 A Review on Multi-Label Learning Algorithms

A Multilabel classification(MLC) problem comes in the form of a training example containing multiple labels associated with to it. Take the VRD problem a pair of objects will

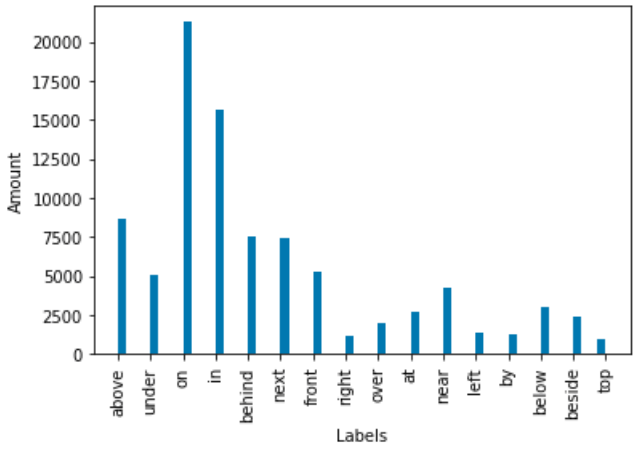

Figure 3: VRD Distribution of Classes

have multiple correct relationships attached to them and sometimes won't have a best descriptor for them making them all equally valid. Therefore it would be best to train it as a MLC problem where multiple correct predictions are correct. Taking the VRD problem as a single label classification (SLC) problem would lead to alienating other correct possible answers.

The main concepts taken from this paper for this dissertation are the evaluation metrics used for evaluating multi-label classifiers. The evaluation metrics can either be Example-based or Label-based. The example-based metrics work by evaluating the example instances separately and then returning the mean value across all of the test data. The label-based metrics are opposite to the one above as they evaluate the systems performance on each class label separately and then return the micro/macro-averaged values across all labels. Example-based Metrics include Subset Accuracy, Hamming Loss, One-error, Average Precision, Coverage, Ranking Loss, Accuracy, Precision, Recall, F^B . The Label-based metrics focus on using the True Positive(TP), False Positive(FP), True Negative(TN) and False Negative(FN) for each label through out the test data. The number of examples is denoted as n, the ground truth label is Y_i and $h(x_i)$ is the predicted label output of the i^{th} example.

Example Based:

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap h(x_i)|}{|h(x_i)|} \tag{1}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap h(x_i)|}{|Y_i|} \tag{2}$$

Label Based:

$$B \in Accuracy, Precision, Recall, F^B \tag{3}$$

Macro-averaging

$$B(h) = \frac{1}{q}\sum_{j=1}^{q} B(TP_j, FP_j, TN_j, FN_j) \tag{4}$$

Micro-averaging

$$B(h) = B(\sum_{j=1}^{q} TP_j, \sum_{j=1}^{q} FP_j, \sum_{j=1}^{q} TN_j, \sum_{j=1}^{q} FN_j) \tag{5}$$

$$Recall(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{(TP_j + FN_j)} \tag{6}$$

$$Precision(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{(TP_j + FP_j)} \tag{7}$$

$$F1(TP_j, FP_j, TN_j, FN_j) = \frac{(1 + B^2) * TP_j}{((1 + B^2) * TP_j + B^2 * FN_j + FP_j)} \tag{8}$$

Recall is described as the intersection of the relevant labels and retrieved labels over the total number of relevant labels. Precision is described as the intersection of the relevant labels and retrieved labels over the total number of retrieved labels. F1-Score is the harmonic average between precision and recall. These metrics are important for the evaluation of MLC models in this dissertation as standard SLC metrics aren't viable.

2.6 Activation Maximization

Once the spatial relations are trained and tested, Activation Maximisation will be used to maximize the outputs of all classes for all the Convolutional Neural Network models. Activation maximisation works by maximising the activation of certain class so that we can see a representation of what the CNN is looking for when classifying a class. This is an easy and useful way of visualizing the final outputs of the CNN for human understanding .

3 Methodology

In this section we will be able to see the implementations needed to be done to reach the goals and objectives of this project.

3.1 Data Preparation

The Stanford VRD and SpatialVoc2k had been used to train the different network architectures and evaluate any differences between them. The dataset had been prepared for two uses, training a Convolutional Neural Network with images and training a Feed Forward Network using geometric features.

3.1.1 Stanford VRD Dataset

This dataset information is stored in the form of dictionaries in two JSON files containing the training and testing data. The data was loaded and the image location, object names, relationships and bounding box locations had been extracted. Firstly the relationships had been quantified and filtered to only include those that are spatial prepositions. Those with synonyms have been combined under one predicate name and those with low amounts of examples hadnt been included. Then the images had been loaded using their image location so that their height and width could been extracted to add to the existing data. Data entries where a pair of objects with same bounding boxes had multiple relationships to them had been concatenated to form one data entry with multiple relationships. Then the final list was randomized and stratified sampled into training/testing/validation data with percentage of 60/20/20. Stratified sampling was done by first distributing all the multi-relation entries and then distributing the single-relation entries as it is easier to fit them and would create a more specific distribution. This was repeated for 10 times to have 10 different training/testing/validation datasets so that multiple models could be trained and their averages taken.

3.1.2 SpatialVoc2k Dataset

This dataset was easier to work with as it already had the image width and height already stored in its JSON file together with the fact that it was specialized for Spatial relations therefore no filtering of spatial relations was necessary apart from removing 3 classes with less than 3 instances each. The data had already multiple relationships grouped for a pair of objects, therefore all that was needed was to randomize it and to apply stratified sampling for training/testing/validation datasets with percentage distributions of 60/20/20, the same process was used as before by first distributing the multiple-relations and then the single-relations. This was repeated for 10 times to have 10 different training/testing/validation datasets so that multiple models could be trained and their averages

taken.

3.1.3 Geometric Datasets

From the previously created datasets more datasets have been created. These datasets contained the geometric features derived from the pair of bounding boxes given to the objects. To preserve the distribution of classes the training/testing/validation data had not been joined together but kept separately, this would enable for the results to be compared fairly. The object labels together with the directions are encoded using One Hot encoding. Note : Let distance from image edge of left and right edges be a1,b1 for first box and a2,b2 for second box the same thing was done for top and bottom edges for c1,d1 and c2,d2.

F0 :	Object Label Ls: Subject
F1 :	Object Label Lo: Object
F2 :	Area of Objs normalized by Union Box Size
F3 :	Area of Objo normalized by Union Box size
F4 :	Ratio of Bounding Box Objs to that of Objo
F5 :	Distance between bounding box centers normalized by Union Box Diagonal
F6 :	Area of Overlap of Bounding Boxes normalized by area of smaller bounding box
F7 :	Minimum Distance between the two bounding boxes
F8 :	Position of Objs relative to Objo in terms of North,South,East,West
F9-F10 :	F9 = (a2-a1)/(b1-a1), F10 = (b2-a1)/(b1-a1)
F11-F12 :	F11 = (c2-c1)/(d1-c1), F12 = (d2-c1)/(d1-c1)
F13 :	Aspect ratio of box Objs
F14 :	Aspect ratio of box Objo
F15 :	Relationship

Table 1: Geometric Features

3.1.4 Single Label Datasets

The datasets that have been created thus far all had multiple relationships between the pair of objects. These datasets would be used to train MLCs therefore for comparison they have been taken as they are and expanded to also train new networks as a SLC problem. Meaning that they were loaded in preserving their distribution and if a data entry had two or more relationships attached to it then it would separate into multiple single-relation entries. This had to be done in this order as if you had used stratified sampling to randomize and distribute the single labels first then there would be entries

with same characteristics with different relationships but in different datasets and it would reduce the amount of possible Multi-Labels.

3.2 Image Preparation

As it has been shown in A Study on the Detection of Visual Relationships (2018) that the best performing method for the VGG16 architecture is the Union-WB-B method. This is where the Union box of the bounding boxes of the pair of objects has been taken, with the background set to Black $[0, 0, 0]$,Subject Bounding Box set to Green $[0, 255, 0]$ and the Object Bounding Box set to Blue $[0, 0, 255]$.

Figure 4: Image from VRD dataset

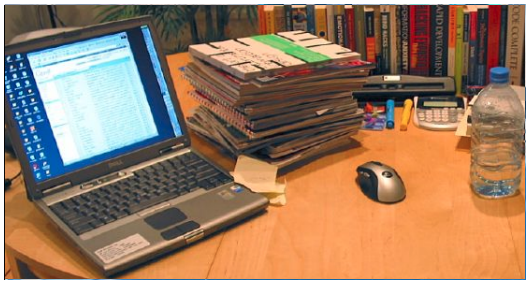

Figure 5: Union Box of $< Laptop, Left - of, Bottle >$

Since this method doesnt use any of the actual objects but only their bounding boxes, there is no need for an actual image only its meta-data. Hence the datasets had been prepared in the form of [Labels, width, height, subject bounding box, object bounding box, subject label, object label]. From this data OpenCv was used create a black image of a certain width, height together with green, blue rectangles added to it in positions of the

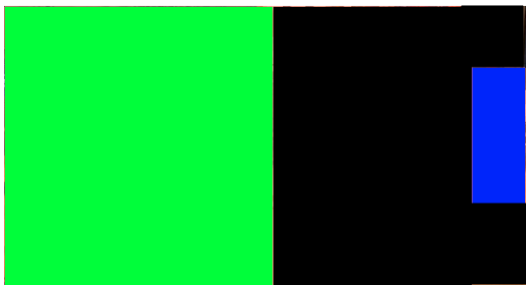

Figure 6: Union-WB-B of $< Laptop, Left-of, Bottle >$

subject/object bounding boxes. This method was much faster as it didnt require any space, processing and loading time for images. The created images are then resized to 224x224 as that is what the VGG16 network had been initially trained using. It is important that during creation the bounding boxes do not overwrite each other in the image but instead if there is an overlap of objects, the overlapping pixel values will be set to $[0, 255, 255]$. This is so that the spatial masks are fed into separate colour channels maintaining their true form.

3.3 Training

A Fine-Tuned VGG16 with ImageNet weights is trained on the datasets and evaluated. A Feed Forward Neural Network was trained on the geometric features exctracted from the bounding boxes of the datasets.

3.3.1 VGG16

The process of Fine-Tuning the VGG16 model started by first loading the model with all its layers and weights. The last Dense layer was replaced with a new Dense layer with a specified amount of classes and activation function that is set before training according to the problem. The MLC problem would use the loss function of binary-crossentropy and an activation function of Sigmoid while the SLC problem would use the categorical-crossentropy loss function and an activation function of Softmax. All the layers had been set to non-trainable except for the last dense and fully connected layers. Using the optimizer stochastic gradient descent(SGD) with a learning of 0.001 and a Nesterov momentum of 0.9 the model was run for 5 Epochs for the VRD dataset and 15 Epochs for SpatialVoc2k dataset. Once the model finished training it was saved and a new model was created with all the layers set to non-trainable except that of the fully connected layers and the last

convolutional block (Conv Block 5). The previous models weights had been loaded into the new model and again run for 5 Epochs for the VRD dataset and 15 Epochs for SpatialVoc2k dataset. Finally the last step was repeated but with a learning rate of 0.00001 for 5 Epochs for the VRD dataset and 15 Epochs for SpatialVoc2k dataset. This was done 10 times for both datasets.

3.3.2 Feed Forward Neural Network

A Feed Forward Neural Network was trained on the Geometric Features created from the datasets. The network had been made up out of two Densly connected layers (256 neurons followed by 128 neurons) and a densly connected output layer containing the number of output classes with an activation function according to the problem being solved. The same parameters were applied to it with regards to MLC and SLC problems. The optimizer ADAM was used with default values and was trained for 10 Epochs. This was done 10 times for both datasets and their results recorded.

3.3.3 Data Generators

Due to having a large data set custom Image generators had been used to load the image meta data and create the images. The batch size for the image generators and CNN had been set to 32. Once a batch of images had been created their pixel intensities had been rescaled to be between 0 and 1, this enables for faster and more precise training of CNNs. The Feed Forward network used custom data generators to load all the geometric features and rescale all the input features to be between 0 and 1. The language features and compass directions had been One Hot Encoded so that they could be processed by the Neural Network.

3.4 Evaluation and Metrics

Once the model had been fully trained an Image Generator had been loaded in with the test data. The model used the predict functionality to predict probabilities on a given image. The predict function was used over the evaluation function as specific MLC evaluation metrics had to be implemented to evaluate the MLC models. On a given image prediction a set of probabilities had been returned corresponding to the probability of each class detected by the model. Since this is a MLC problem the Sigmoid activation function was used so each class had their own independent probability ranging from 0% to 100%. A

threshold of 50% had been chosen as a probability cut off point. If a value was above 50% then it would be turned on (set to 1) and if it was below then it would have been turned off (set to 0). The predicted values had then been compared to the ground truth labels and documented per class. The True positive, False positive , True negative and False negative values for each class are decided with the use of the help of the contingency table in Figure.7.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 7: Contingency table of Ground Truth vs Predicted Values

Once predictions have been made on the testing data and values recorded, the evaluation metrics taken from "A Review on Multi-Label Learning Algorithms" mentioned above were run and recorded. These metrics are used to evaluate MLC models. To evaluate SLC models Recall@1,Precision@1 and F1-Score@1 had been utelized. Since @1 is used it means that the highest probability value is used for the metric therefore the predictions were first ranked in descending order by probability and the highest predicted class is compared to the ground truth value. The results are recorded per class together with the Micro/Macro averages. All 10 models for each problem had been evaluated and their averages been tabulated.

3.5 Interpreting the models

To interpret the models via activation maximization the trained models are first loaded in, then the last activation function of the fully connected prediction layer was chosen and replaced by a linear activation. Activation maximization was then initialized on a random input image and run for 1024 back propagation iterations maximizing the output for each class. The resulting images were then saved and compared for each model. It was made sure that the input range was set to between 0 and 1 as that is scale the CNN's had been trained on.

4 Findings

4.1 Preamble

In this chapter the results achieved from various models trained in Chapter 3 are presented. Every dataset shows the results achieved on the test data for both the VGG16 and Feed Forward Neural Network together MLC vs SLC training methods.

4.2 VRD Dataset Evaluation Results

In this section the results evaluated from all the models trained on the VRD dataset are presented.

4.2.1 VGG16 Evaluation Results

This section shows the relevant results obtained by models for both MLC and SLC problems.The presented results are rounded to 2 decimal places.

Class	MLC	MLC	MLC	SLC	SLC	SLC
English	Precision	Recall	F1	Precision@1	Recall@1	F1@1
above	0.66	0.44	0.53	0.52	0.63	0.57
at	0.00	0.00	0.00	0.11	0	0
behind	0.56	0.03	0.06	0.31	0.26	0.28
below	0.38	0.03	0.06	0.35	0.22	0.27
beside	0.00	0.00	0.00	0.03	0	0
front	0.00	0.00	0.00	0.25	0.21	0.23
in	0.79	0.48	0.59	0.65	0.55	0.6
left	0.13	0.00	0.01	0.3	0.22	0.25
near	0.00	0.00	0.00	0.19	0.03	0.05
next	0.44	0.02	0.04	0.3	0.44	0.36
on	0.66	0.69	0.67	0.56	0.86	0.68
right	0.10	0.00	0.00	0.24	0.2	0.22
under	0.48	0.05	0.09	0.33	0.45	0.38

Table 2: VGG16(VRD) results for MLC and SLC

Class	MLC	MLC	MLC	SLC	SLC	SLC
Metric Type	Precision	Recall	F1	Precision@1	Recall@1	F1@1
Macro-Average	0.26	0.11	0.13	0.26	0.26	0.24
Micro-Average	0.68	0.29	0.41	0.47	0.47	0.47
Example	0.31	0.31	0.31			

Table 3: Example/Micro/Macro Average Label Based Metrics for VGG16(VRD)

The results in Table 2 do not include the following labels:={ by, over, top} as they have achieved a result of 0 for all metrics. These results show the averages of 10 VGG16 models Fine-Tuned using the VRD dataset as MLC and SLC problems. The Example based metric as seen in the MLC problem is not included for the SLC problem as it produces equivalent results to that of the Micro-Average Metric. The results shown for @1 mean only the top 1 predicted labels are compared to the ground truth.

4.2.2 Feed Forward Evaluation Results

This section shows the relevant results obtained by the Feed Forward models for both MLC and SLC problems.The presented results are rounded to 2 decimal places.

Class	MLC	MLC	MLC	SLC	SLC	SLC
English	Precision	Recall	F1	Precision@1	Recall@1	F1@1
above	0.66	0.62	0.64	0.57	0.64	0.60
at	0.26	0.11	0.16	0.19	0.13	0.15
behind	0.56	0.41	0.47	0.46	0.49	0.48
below	0.39	0.24	0.29	0.33	0.25	0.28
beside	0.20	0.04	0.07	0.13	0.07	0.09
by	0.23	0.05	0.07	0.12	0.07	0.09
front	0.43	0.26	0.32	0.34	0.34	0.34
in	0.80	0.68	0.73	0.72	0.70	0.71
left	0.31	0.12	0.17	0.24	0.19	0.21
near	0.22	0.07	0.10	0.17	0.13	0.15
next	0.35	0.21	0.26	0.29	0.32	0.30
on	0.77	0.76	0.76	0.70	0.77	0.73
over	0.30	0.09	0.14	0.24	0.11	0.15
right	0.27	0.12	0.16	0.23	0.19	0.20
top	0.18	0.03	0.06	0.12	0.05	0.06
under	0.54	0.50	0.51	0.43	0.54	0.48

Table 4: Feed Forward NN(VRD) results for MLC and SLC

Class	MLC	MLC	MLC	SLC	SLC	SLC
Metric Type	Precision	Recall	F1	Precision@1	Recall@1	F1@1
Macro-Average	0.40	0.27	0.31	0.33	0.31	0.31
Micro-Average	0.63	0.47	0.54	0.51	0.51	0.51
Example	0.49	0.49	0.49			

Table 5: Example/Micro/Macro Average Label Based Metrics for Feed Forward NN(VRD) These results show the averages of 10 Feed Forward Neural Network models trained using VRD Geometric Features as MLC and SLC problems. The Example based metric as seen in the MLC problem is not included for the SLC problem as it produces equivalent results to that of the Micro-Average Metric. The results shown for @1 mean only the top 1 predicted labels are compared to the ground truth.

4.2.3 Activation Maximization Evaluation Results

4.3 SpatialVoc2k Dataset Evaluation Results

In this section the results evaluated from all the models trained on the SpatialVoc2k dataset are presented.

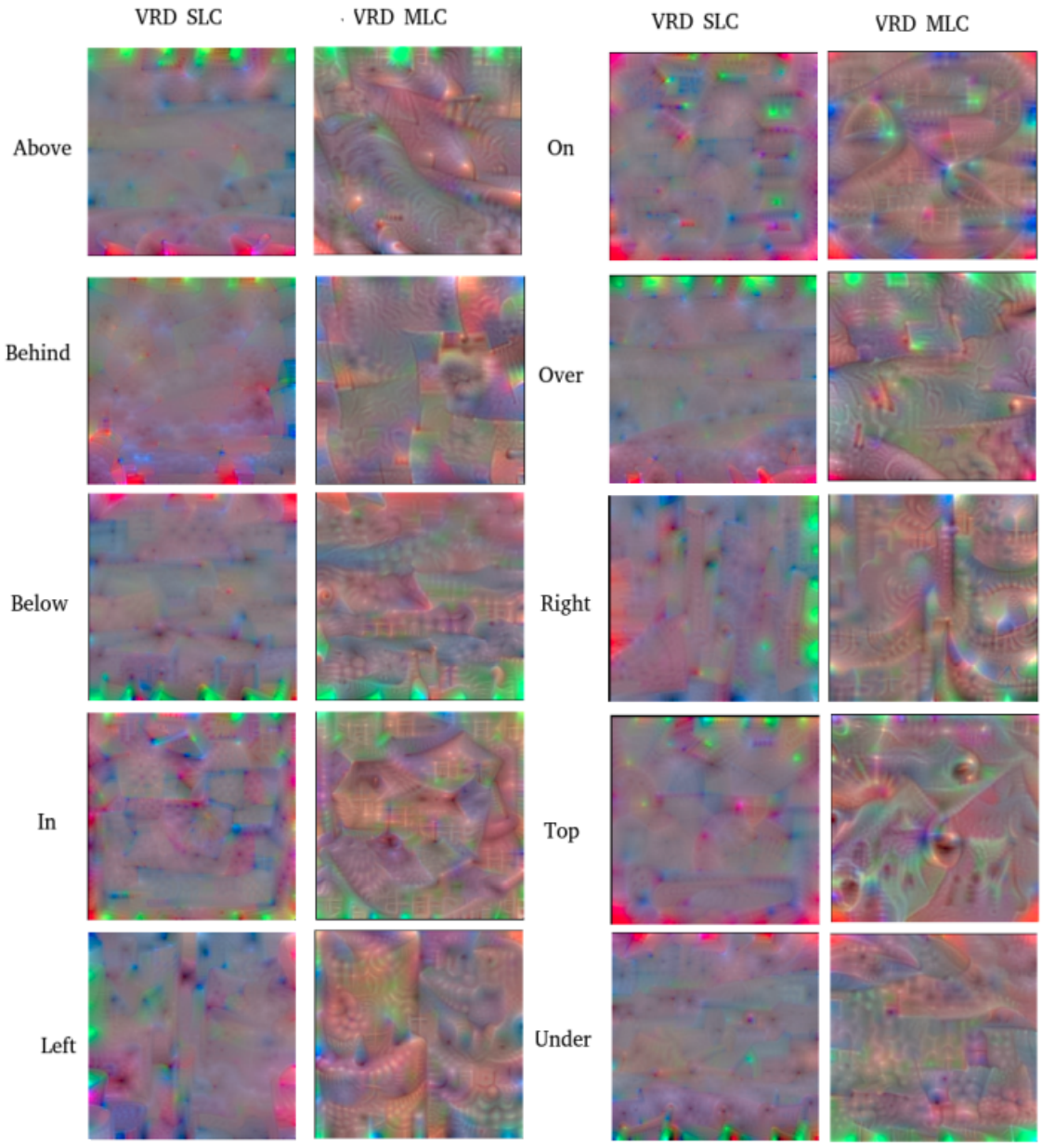

Figure 8: Activation Maximization on VGG16(VRD) Classes for MLC and SLC

4.3.1 VGG16 Evaluation Results

Class	Class	MLC	MLC	MLC	SLC	SLC	SLC
French	English	Precision	Recall	F1	Precision@1	Recall@1	F1@1
a cote de	next to	0.67	0.52	0.58	0.27	0.20	0.23
a l'exterieur de	outside	0.00	0.00	0.00	0.03	0.01	0.01
au dessus de	beyond	0.00	0.00	0.00	0.15	0.09	0.11
au niveau de	near/at the level of	0.68	0.47	0.56	0.30	0.17	0.22
aucun	none	0.00	0.00	0.00	0.10	0.02	0.03
autour de	around	0.00	0.00	0.00	0.47	0.31	0.32
contre	against	0.51	0.14	0.21	0.22	0.16	0.18
dans	in/into	0.00	0.00	0.00	0.37	0.28	0.30
derriere	behind	0.63	0.42	0.50	0.28	0.34	0.30
devant	front/before	0.64	0.46	0.53	0.30	0.34	0.32
en face de	across from	0.00	0.00	0.00	0.07	0.02	0.03
loin de	far from	0.70	0.25	0.36	0.37	0.30	0.33
pres de	near/by	0.75	0.72	0.73	0.32	0.44	0.37
sous	under/below	0.64	0.41	0.50	0.34	0.48	0.40
sur	on	0.67	0.45	0.53	0.36	0.46	0.40

Table 6: VGG16(SpatialVoc2k) results for MLC and SLC

Class	MLC	MLC	MLC	SLC	SLC	SLC
Metric Type	Precision	Recall	F1	Precision@1	Recall@1	F1@1
Macro-Average	0.35	0.23	0.27	0.23	0.21	0.21
Micro-Average	0.69	0.47	0.55	0.30	0.30	0.30
Example	0.58	0.46	0.51			

Table 7: Example/Micro/Macro Average Label Based Metrics for VGG16(SpatialVoc2k) The results in Table 6 do not include the following labels:={ par dela(beyond), le long de(along)} as they have achieved a result of 0 for all metrics. These results show the averages of 10 VGG16 models Fine-Tuned using the SpatialVoc2k dataset as MLC and SLC problems. The Example based metric as seen in the MLC problem is not included for the SLC problem as it produces equivalent results to that of the Micro-Average Metric. The results shown for @1 mean only the top 1 predicted labels are compared to the ground truth.

4.3.2 Feed Forward Evaluation Results

In this section the results for the Feed Forward Neural Networks are presented trained on the SpatialVoc2k dataset as MLC and SLC problems.

Class	Class	MLC	MLC	MLC	SLC	SLC	SLC
French	English	Precision	Recall	F1	Precision@1	Recall@1	F1@1
a cote de	next to	0.64	0.57	0.60	0.25	0.11	0.14
a l'exterieur de	outside	0.00	0.00	0.00	0.10	0.07	0.07
au dessus d e	beyond	0.00	0.00	0.00	0.18	0.03	0.04
au niveau de	near/at the level of	0.62	0.39	0.47	0.24	0.02	0.04
autour de	around	0.00	0.00	0.00	0.58	0.44	0.44
contre	against	0.51	0.25	0.33	0.24	0.10	0.14
dans	in/into	0.17	0.03	0.04	0.49	0.33	0.36
derriere	behind	0.71	0.41	0.52	0.35	0.37	0.35
devant	front/before	0.67	0.43	0.52	0.36	0.34	0.35
en face de	across from	0.17	0.01	0.03	0.23	0.08	0.11
le long de	along/by	0.00	0.00	0.00	0.03	0.01	0.01
loin de	far from	0.69	0.36	0.46	0.36	0.22	0.26
pres de	near/by	0.76	0.79	0.78	0.32	0.64	0.43
sous	under/below	0.78	0.65	0.70	0.50	0.66	0.57
sur	on	0.79	0.75	0.76	0.48	0.73	0.58

Table 8: Feed Forward NN(SpatialVoc2k) results for MLC and SLC

Class	MLC	MLC	MLC	SLC	SLC	SLC
Metric Type	Precision	Recall	F1	Precision@1	Recall@1	F1@1
Macro-Average	0.38	0.27	0.31	0.29	0.25	0.23
Micro-Average	0.70	0.51	0.59	0.34	0.34	0.34
Example	0.64	0.52	0.57			

Table 9: Example/Micro/Macro Average Label Based Metrics for Feed Forward NN(SpatialVoc2k)

The results in Table 10 does not include the following label:={ aucun(none)} as it has achieved a result of 0 for all metrics. These results show the averages of 10 Feed Forward NN models trained using the SpatialVoc2k dataset as MLC and SLC problems. The Example based metric as seen in the MLC problem is not included for the SLC problem as it produces equivalent results to that of the Micro-Average Metric. The results shown for @1 mean only the top 1 predicted labels are compared to the ground truth.

4.3.3 Activation Maximization Evaluation Results

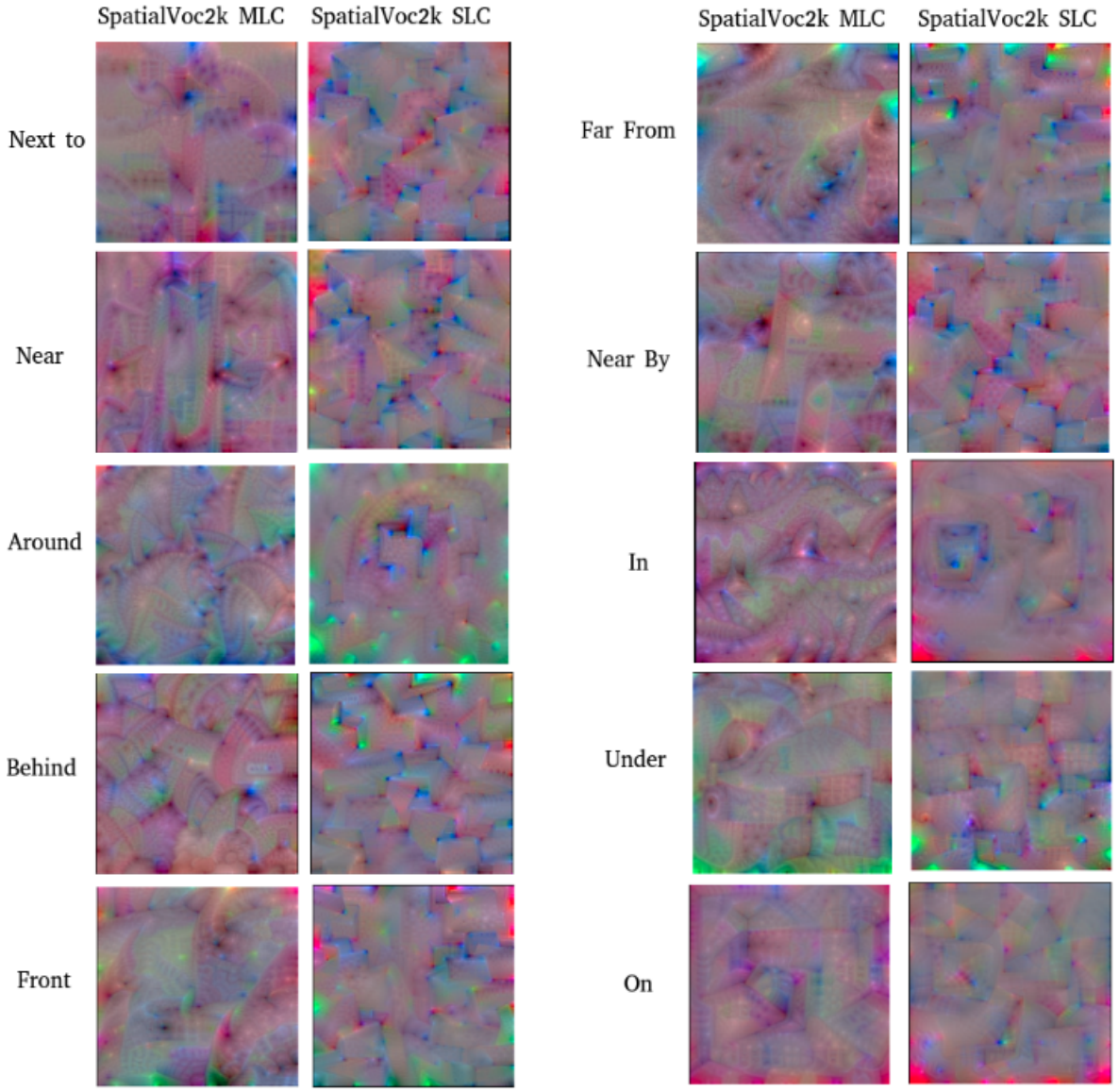

Figure 9: Activation Maximization on VGG16(SpatialVoc2k) Classes for MLC and SLC

5 Analysis of Results

5.1 Preamble

In this chapter the results obtained in Chapter 3 are discussed. This chapter is split into three parts,-Section 5.2 analyses and compares the MLC vs SLC problems, VGG16 vs NN and the Activation Maps for the VRD dataset. -Section 5.3 tackles the same results and problems but for the SpatialVoc2k dataset instead.

5.2 VRD Results

In this section the results obtained by the VRD dataset will be discussed and compared across models and training methods.

5.2.1 Multi-label vs Single-label Classification

The VGG16 Single-Label Classification training method outperformed the Multi-Label Classification training method for all per-predicate accounts of Recall and F1. The MLC Precision metric recorded better results for labels:={ above, behind, below, in, next, on, under} compared to that of SLC. The Macro-Average, meaning the average of all the predicates combined achieved equal results for precision and higher results in Recall/F1 for the SLC method. This means that even though the MLC training method returned less predicate results the ones it did return had been more relevant compared to that of the SLC. SLC had higher or equal Recall for all predicates compared to MLC meaning that most of the relevant labels had been retrieved for SLC compared to that of MLC. SLC has more predicate results due to the difference in metrics, MLC metrics have a threshold of 50% which filter out low probability labels, while SLC uses @1 so low probability labels can be retrieved as long as they are the highest among the predicted.
The Feed Forward Neural Network had a greater MLC precision score than that of the SLC for all accounts of per-predicate labels while SLC had better results for recall. Even though SLC produced higher recall for all predicates the results aren't far off from MLC, the greater difference is in the precision scores. SLCs Micro-Average score shows 51% for all metrics while MLC's Micro-Average score shows Precision: 63%, Recall: 47%, F1: 54%. The difference in Micro-Average Precision made the MLC method outperform in F1 score over the SLC method.
The MLC method proved to be more precise than the SLC while SLC was more sensitive

and had recalled the most relevant labels.

5.2.2 VGG16 vs Feed Forward Neural Network

The Feed Forward Neural Network trained on the Geometric Features achieved better results than the Fine-Tuned VGG16 CNN trained on spatial masks. The geometric features didn't only contain spatial configurations of the bounding boxes but also the subject/object categories which the spatial masks couldn't possibly learn. These language features that have been combined with the geometric features are the main reason that the Feed Forward Network outperforms the VGG16. Language features allow for more ambiguous predicates to be predicted as these predicates would be indistinguishable as spatial masks or geometric features alone but would have a very distinct occurance between a pair of object categories. To show the importance of the language features in the appendix there are results of the Feed Forward Neural Network trained using only Geometric Features meaning that the subject/object categories have been removed. These results show a lower performance in all metrics than that of a Fine-Tuned VGG16 for both MLC and SLC. They also show that SLC has a higher F1 score than MLC which is the opposite of what was shown in the networks trained with both Language and Geometric features.
The results achieved by the Feed Forward Neural Network using Geometric and Language Features outperform VGG16 for all metrics. For real life implementation I would suggest using CNN's for object detection and localization from there extract the Geometric and Language features from the pair of objects and train a Feed Forward Neural Network. The Feed Forward Neural Network takes less time to train and produces better results. To improve these results word2vec from Lu et al(2016) should be used on the language features instead of One Hot Encoding, this would enable for even faster training as you won't have a large input shape for all the possible object categories and increased results on unseen category combinations.

5.2.3 Activation Maximization

The Activation Maps show what the CNN is looking for, for that specific output class. Knowing that the subject is Green and object is Blue we can interpret these maps. The predicates Above,Over have similar activation maps while Top has a less clear version of those maps. These maps show that Green is Above/Over/Top the Blue colour which is a good indicator of how the network perceives these spatial relations. The below and under activation maps show the Green colour below/under the Blue colour which is a straight

forward indicator. Left and Right also have quite clear activation maps that are easily understandable. The predicate "In" is shown to be a mix of $[0, 255, 255]$ pixels surrounded by an outline of Blue pixels indicating that the Green subject is inside the Blue object. "On" has multiple small concentrations of Green points above small concentrations of $[0, 255, 255]$ followed by Blue points, this indicates that the predicate "On" has a Green subject constantly in contact with the Blue object while being in a higher position over it at the same time. While "above/over/top" predicates have the concentration of Green confined to the upper limits of the activation map "On" has them more spread out. The MLC activations produced similar looking maps to that of the SLC activations but they are less clear.

5.2.4 Conclusions

These results dictate that even though the Visual Relationship Detection problem should be taken as a Multi-Label Classification problem the label distributions in the datasets play a large role in how well the model will be able to recognise multiple relationships. Training a model which is heavily composed of single labels(VRD) as a Multi-Label model hinders the models ability to predict labels correctly.
The Feed Forward Neural Network outperforms the CNN mainly due to the Language Features otherwise having only Geometric Features produce lower scores than the CNN. Language features also greatly close the gap in results between MLC and SLC for the VRD dataset.

5.3 SpatialVoc2k Results

In this section the results obtained by the SpatialVoc2k dataset will be discussed and compared across models and training methods

5.3.1 Multi-label vs Single-label Classification

SpatialVoc2k is mainly a multi-label model and it proves that in MLC's results compared to that of the SLC training method.The VGG16 MLC models scores higher for all the evaluation metrics in the Macro and Micro-averages. All the non-zero per-predicate results returned by the MLC model had higher score than that of SLC except for the predicate "far from" which had a higher recall for SLC method. The zero per predicate results returned by the MLC don't necessarily show that the MLC model doesn't detect them but

that their probabilities weren't high enough to be considered as correct. The SLC method has noteably higher results on the predicates "around/into" compared to the zero results retrieved by MLC, this can be explained further by looking at the activation maps.
NOTES Feed Forward Neural Network exhibits the same MLC vs SLC properties as VGG16

5.3.2 VGG16 vs Feed Forward Neural Network

NOTES Feed Forward Neural Network outperforms the VGG16 in all aspects largely due to the language features NOTES Geometric Features alone performed worse than the VGG16 fine-tuned for all metrics and accounts.

5.3.3 Activation Maximization

NOTES MLC has clearer maps than SLC except on "Around" and "In" where SLC has had higher metric scores shows how activation maximization can be used to understand these neural networks

5.3.4 Conclusions

Acknowledgements.

I would like to thank and express my special gratitude to my supervisor Dr. Adrian Muscat for assisting me and guiding me throughout this final year project .

6 References

TO DO ADD REFERENCES Add a terminology page with regards to training epoch/loss/optimizers Add paragraph about SmallVGG Add appendix with SmallVGG results together with Geometric Only Results without language features. Literature Review of Activation Maximization