2022

KG-CNN: Augmenting Convolutional Neural Networks with Knowledge Graphs for Multi-Class Image Classification

Aidan O'Neill
Technological University Dublin, Ireland

KG-CNN: Augmenting Convolutional Neural Networks with Knowledge Graphs for Multi-class image classification

Aiden O'Neill

A dissertation submitted in partial fulfilment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computing (Data Science)

June 2022

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Science), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: *Aiden O'Neill.*

Date: 16/06/2022

Abstract

Computer vision is slowly becoming more and more prevalent in daily life. Tesla has recently announced that it plans to scale up the manufacturing of their Robotaxis by 2024, with this increase in self-driving vehicles being just one example, the importance of computer vision is growing year by year.

Vision can be easy to take for granted, as most humans grow up using vision as their primary way of absorbing environmental information. The way humans process and classify visual information differs significantly from how current computer vision systems process and organise visual information. The human brain can use its past knowledge and experience to draw conclusions that would be impossible for a limited artificial system. The human brain has a lifetime of visually informed learning to fall back on when classifying an object, the function of an object, and the relationships between particular objects. Humans can recall this information when needed once initially learned. Current convolutional neural networks (CNN) focus on processing the image at a pixel level, so they do not have this historical and relational information about these objects to fall back on, at least not explicitly. A CNN may connect certain features or objects during the classification process but this is a black box, and the end user has now way of determining this links.

In this paper, a CNN is used to classify standard household rooms; it has been trained on the room images and has no information about the various objects that commonly appear in each room. Knowledge graphs augment the CNN output to fill the absence of this historical and relational information, it can also provide some level of explainability to the system. Object recognition is used on a distinct subset of the dataset to build these knowledge graphs, with the item label and the frequency of occurrence recorded for each item. Once generated, these graphs can then augment the low confidence classification outputs from the CNN. When an image is classified by the CNN, if it falls below a certain level of confidence, the image will go through the object detection process and the results will be mapped to a knowledge graph.

This knowledge graph can then be compared against the knowledge graphs of each room, using a naive Bayes classifier to generate the likely probabilities that the objects appear in a given room. These probabilities are then used to revise or reinforce the image classification label as appropriate.

Keywords: image classification, knowledge graph, computer vision, object recognition

Acknowledgments

Sincere thanks to my supervisor, Jack O'Neill. I am genuinely grateful for the continuous support, guidance and advice over the last six months. You enabled me to push this study passed where I thought possible. I also wish to thank my friends and family for their continuous support over the last two years. Finally, I would like to thank my employer and, more directly, my manager, who has supported and enabled me to undertake this degree.

Contents

List of Figures

List of Tables

List of Acronyms

CNN Convolutional Neural network

KG Knowledge Graph

OR Object Recognition

AI Artificial Intelligence

ML Machine Learning

ReLU Rectified Linear Activation Unit

NLP Natural Language Processing

DCNN Deep Convolutional Neural Network

RCNN Region-Based Convolutional Neural Network

MPCNN Multi-Part Convolutional Neural Network

QCNN Quantum Convolutional Neural Network

ILSVRC ImageNet Large Scale Visual Recognition Challenge

COCO Common Objects in Context

CIFAR Canadian Institute for Advanced Research

LVIS Large Vocabulary Instance Segmentation

Chapter 1

Introduction

This section will discuss the study's background scope and summarise the research question, objectives and methodologies before outlining the scope and limitations of the experiment.

1.1 Background

As the prevalence of the internet and social media continues to grow, an ever-increasing number of images are being uploaded every year. According to the Instagram usage stats for 2022, 100 million images are uploaded to their platform alone every day[1]. This increase in the volume of images captured and uploaded has provided an almost infinite data source for the field of computer vision. One of the key sub-disciplines in this field is image classification; this deals with how a system can analyse an image and determine the object it represents. This has a wide range of real-world applications, including medical imaging, traffic management systems, and object detection for satellite imagery.

Convolutional neural networks (CNN) are the most widely used deep learning methods for image classification and, as such, have been shown to achieve a high degree of accuracy. CNNs are scalable to large datasets and reduce the need for feature engineering in more traditional algorithm-based image classifiers. The convolutional layers, which give CNNs their name, allow the neural network to extract both local and global features from the image and help increase performance. CNNs are not perfect; however, if using a small dataset, the level of bias introduced from the training set can cause significant fluctuations in accuracy. Light intensity, seen in Finlayson (2018),

[1]https://www.omnicoreagency.com/instagram-statistics/

object orientation, shown by Brown *et al.* (2021), and object overlap, explored by Paulauskaite-Taraseviciene *et al.* (2019) are some of the issues that can cause the introduction of bias into the trained model. Ensuring that the dataset accurately represents real-world conditions is a challenge that ideally needs both a large enough dataset and image augmentation techniques to simulate these varying conditions, seen in the application of image retrieval by Dharani & Laurence Aroquiaraj (2017). CNNs are also a black box type of system; as such, there is no real way to explain why a model classifies an image as it does; some inferences can be made on this by examining the layers, as shown by Pellegrini (2016), but there is no way to be specific.

Knowledge graphs, also called semantic networks or semantic webs, are a way to visually represent a network of entities such as objects, events, or concepts, and show the relationship between these entities. This data is usually represented using a graphical structure which has lead to the adoption of the term knowledge graph. These types of graphs can be very useful in reducing the complexity of the relationships between entities. A classic example of this would be a family tree; with the family members being the nodes of the graph, and their relationships with each other the lines, also called edges, joining these nodes.

This study proposes the addition of a knowledge graph of semantic data gathered from image analysis to a CNN to increase accuracy and provide some level of explainability for the classification label if necessary. A CNN is a feed-forward network that extracts features from images, and it passes these features through the various layers until a classification output is reached. At this stage, adding a knowledge graph can hopefully help explain the relationships of the objects found within the image and either reinforce the CNNs classification label or revise it.

1.2 Research Project/problem

Convolutional neural networks are the most widely used deep learning models for image classification and object recognition, but these have some limitations. A trained neural network has no explicit knowledge of the relationships of the individual objects

that make up the image as a whole, focusing on the tasks of either image classification or object detection and localisation by extracting features. Suppose the relationships between these features can be predetermined and stored as a knowledge base. In that case, it could help potentially provide insight for future trained models and provide higher classification accuracy with less training. This study seeks to analyse this potential and investigate if adding such a knowledge base can help improve classification performance. In this investigation, this increase in performance is measured, and if found, the level of data needed to get this increase can be explored.

This research problem is defined as *"Can the addition of a knowledge graph of image-based semantic data to a convolutional neural network improve the model's accuracy at classifying rooms?"*

1.2.1 Hypothesis

Null Hypothesis (H0): If a convolutional neural network is developed using household image data and augmented with a knowledge graph of component household item semantic data, then it will show no statistically significant increase in accuracy when classifying the image than a conventional CNN model.

Alternate Hypothesis (Ha): If a convolutional neural network is developed using household image data and augmented with a knowledge graph of household item semantic data, then it will perform more accurately when classifying the image than a conventional CNN model.

The hypothesis will be tested using an independent samples t-test to determine if the results are statistically significant with an $\alpha = 0.05$.

1.3 Research Objectives

This experiment aims to determine if adding an external knowledge base can help enhance the performance of a convolutional neural network for the task of image classification, as well as provide some level of explainability for the classification decision. A detailed comparison of two related CNN models, one baseline model and one en-

hanced with a knowledge graph containing semantic data garnered from image data. The ultimate objective is to see if this external knowledge graph can be expanded and used to augment other smaller image classification models. This can be broken down into 3 key research elements:

- Creation of Knowledge Graph - The knowledge graphs are one of the core aspects of this experiment and need to be created from scratch, using a combination of the python libraries NetworkX and Pandas.
- Combination of Knowledge Graph and CNN - This integration is the other core aspect of this experiment, this will explore the best way to combine the output of the CNN and the knowledge graph. With some thresholds to determine if the experiment should rely on the classification from the CNN or make use of the KG
- Design and run the experiment - This final objective is to implement the above two objectives into a practical application that can be run repeatedly. This will make use of the Google Colab as the development environment.

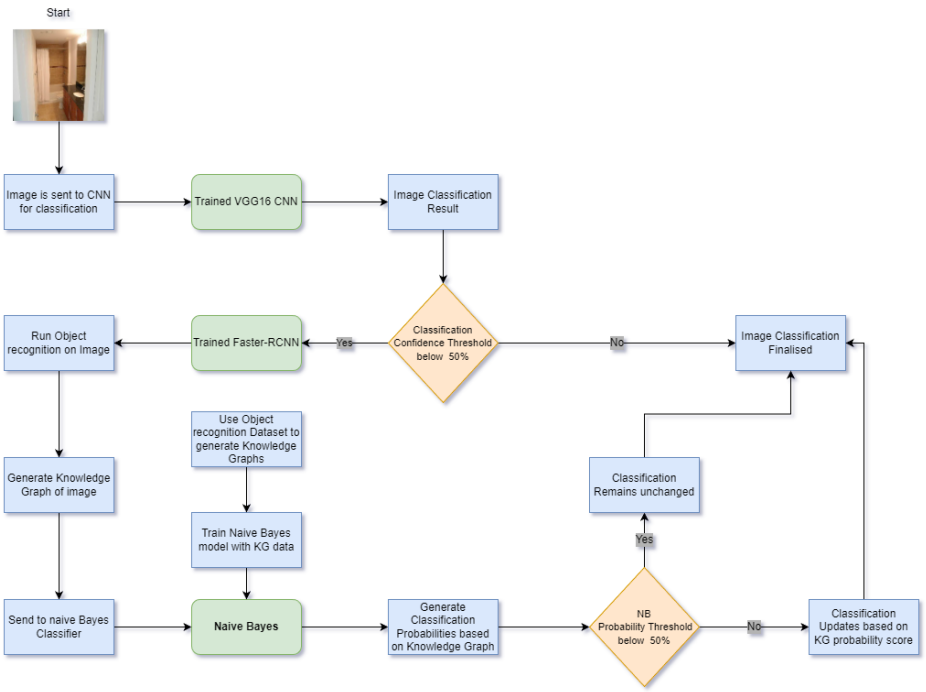

Figure 1.1: High Level system flowchart

Figure 1.1 shows a high level flow chart that details the final planned system, this does not show the training and testing cycles needed for both the VGG16 model used for image classification and the Faster-RCNN model used for object recognition but shows the major decision points within the system and how the CNN and Knowledge graph based Naive Bayes classifier create an ensamble classifier given certain decision points.

1.4 Research Methodologies

This study will use secondary research with a quantifiable approach and deductive reasoning, defining the hypothesis first and then making empirical observations to assess the results and test the hypothesis. It will be an extension of the existing knowledge in the published literature, particularly related to the recent advancements in image classification with convolutional neural networks.

Three separate data sets will be used for this study, containing approximately 9800 images for training with 1200 for testing and 12000 images for object recognition to build a knowledge graph containing images from various domestic household rooms. These datasets were provided from various sources, by MIT[2], Kaggle[3] and ImageNet[4]. Each of the six classes were balanced as much as possible, using roughly the same number of training images to try and remove any potential bias. These datasets are preexisting and were not gathered for this study, so this is secondary research.

This research prioritises accuracy to evaluate model performance and gauge if the addition of the knowledge graph has a quantifiable effect. This study uses this data to answer a specific question, which is classified as empirical research, testing a hypothesis by experiment. This is inductive reasoning, going from a specific use case, in this example, to classify the function of a room to potentially a more general model that can help with any image classification task.

[2]https://web.mit.edu/torralba/www/indoor.html

[3]https://www.kaggle.com/datasets/robinreni/house-rooms-image-dataset

[4]https://image-net.org/download.php

1.5 Scope and Limitations

The scope of this study is to use an external knowledge graph of image-based semantic data to aid in the classification of domestic household locations. Three datasets are used in this study, the first being the Indoor Scene Recognition dataset, created in 2009 and provided by the Massachusetts Institute of Technology (MIT); the second, the House Rooms Image Dataset, is an openly available image set on the data science website Kaggle, uploaded in 2020, and the final dataset in the ImageNet data set, first created in 2006 but has been continually updated to as recently as 2020, this is provided by the Stanford vision and learning lab. The indoor scene recognition and ImageNet datasets are much larger than the scope of this study, so a subset of these are used, using the six classes relevant to this study. The datasets used in this study do not have a ready-made application to substantially impact a practical field, unlike other datasets such as medical imaging or weather satellite images. This can be seen as a limitation in scope. This research will only focus on 1 CNN architecture; the VGG16 architecture will be the foundation for both the baseline and KG enhanced models. The comparison and contrast of the performance of various other CNN models will not be considered for this study.

The research is limited to the six classes chosen, domestic household locations, so any untrained classes will not be in scope for classification. Also, Processing an image is a resource-intensive task and needs more significant clusters of GPUs, TPUs and time for training and testing.

1.6 Document Outline

The following research paper is structured as follows.

- **Chapter 2 - Literature Review**: This chapter provides an overview of the various elements in this study, Convolutional Neural Networks, Knowledge graphs and Object recognition, as well as state-of-the-art innovations in the field of computer vision, with an emphasis on the integration of knowledge graphs to

convolutional neural networks.

- **Chapter 3 - Experiment Design and Methodology**: This chapter summarises the scientific approach undertaken in experimental design and methodology. It discusses all major steps taken to complete the study, including data set description and preparation, neural network selection for both image classification and object recognition tasks, training of the identified neural network models for image classification and object recognition, and each model's architecture and strengths. It also details the generation of a knowledge graph and the integration of all of the individual pieces to generate a quantifiable result set.

- **Chapter 4 - Results, Evaluation and Discussion**: This chapter shows the results from the experimental process detailed in chapter 3, covering each of the different model's results. It also reports the results of statistical tests used to determine if the null hypothesis for this study should be accepted or rejected, as well as any comparisons done, such as the differences in confusion matrices for different knowledge graphs, using appropriate statistical measures.

- **Chapter 5 - Conclusion**: This chapter summarises the whole study, from problem definition to results, with a high-level overview for each section. It also discusses potential future work and developments using this study as a foundation.

Chapter 2

Review of existing literature

This section will explain this paper's core elements, image classification using convolutional neural networks (CNN), object recognition (OR), and knowledge graphs (KG), starting from a basic level to more complex implementations for each. Computational power has become more accessible in recent years with the advances in Graphical Processing units and cloud computing solutions. This increased access to computational power has removed entry barriers for more significant and complex computer vision problems. This section will also highlight the gaps in the research this paper aims to fill within these topics.

2.1 Computer Vision

The core of computer vision is how computers view the world, whether through static images or live real-world data. The most fundamental tasks with this data are image classification and object detection. Image classification and object detection are closely related to computer vision. Image classification can be used to classify individual objects within an image or to classify the image as a whole. Object detection is used to identify and locate multiple objects within an image.

As computer vision is becoming more and more relevant to everyday life, this has led to image classification and object recognition techniques advancing significantly in recent years, as shown by Rashu *et al.* (2021) and Frei & Kruis (2021). Both focus on real-world applications, Rashu *et al.* (2021) use image classification to aid in road safety and governance by removing some of the manual work needed to label relevant vehicles. It is classifying a vehicle by its function, not its shape and size, as has been the traditional way of classifying a vehicle. They managed to achieve a successful

classification rate of greater than 80%. This image classification application can have many real-world applications, such as increasing efficiency at toll booths or expediting the travel of emergency service vehicles in real-time. Frei & Kruis (2021) also look at a real-world application, using object detection to identify fibre-based materials such as asbestos or carbon nanotubes.

Computer vision has evolved from a field that historically relied heavily on human intelligence to a field that now pushes the limitations of Deep learning, "Advances in Computer Vision" (2020) shows that Deep learning techniques have made the historic techniques used for computer vision mostly irrelevant. They note that this knowledge should not be discarded however, having a knowledge of the past techniques can help the field of computer vision as a whole to become more rounded and in some situations the classic computer vision techniques are more suited to problems that do not need the computationally heavy deep learning methods.

2.1.1 Image Classification

Neural networks are this study's main deep learning method for image classification. This section will explain the basics of neural networks and how they have become the go-to method for image classification.

- Feed-Forward neural networks. This type of neural network is the most basic, being the first neural network devised, as shown by Rumelhart *et al.* (1985), with a paper dated from the mid 1980s. Figure 2.1 shows a basic feed-forward neural network. The input is passed from one layer to the next, multiplied by the trained weights in the hidden layer before being sent to the output layer for classification. The values are fed forward from one layer to the next, hence the name.

- Back Propagation is a training algorithm widely used for training feed-forward neural networks. It has a feed-forward element but calculates the error and passes it back to adjust the weights. It does this calculation using gradient descent and an error function. Gradient descent is an optimisation algorithm

and is the most common way to optimise any neural network and helps these models to learn over time, with the goal being to minimise the error/loss in the model, as shown in Figure 2.2. This allows backpropagation to tweak the neural network, improve accuracy, and reduce error on every iteration; generally, there is some function to halt the training process early if no progress is being made, as this indicates that the model may have hit a minimum.

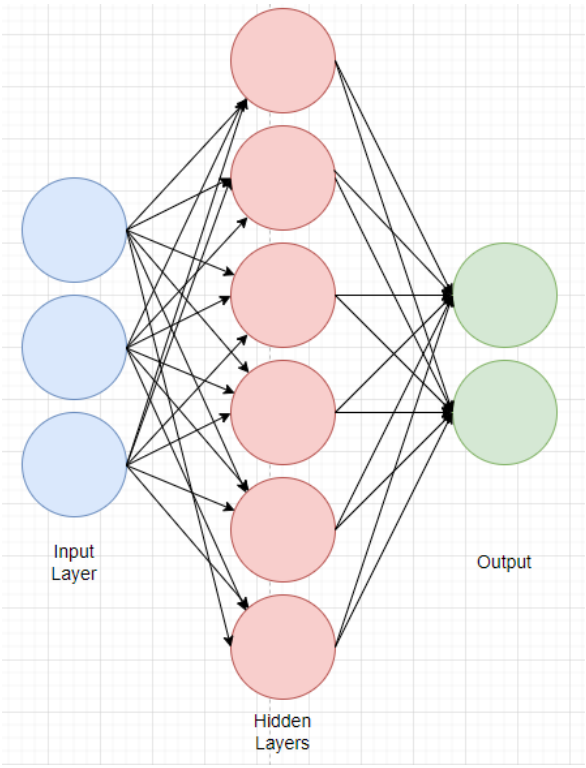

Figure 2.1: Basic Architecture of a Feed-Forward Neural Network

Figure 2.1 shows a basic feed-forward neural network, in this case flowing from the leftmost input layer through the hidden layer, to perform computations and transfer information from the input layer to the rightmost output layer.

Figure 2.2 shows an example of gradient descent; this is used to optimise the algorithm and find the best balance between cost and output. This cost is a way to describe the error in the application, the goal is to get the most accurate output with minimal cost, although in real-world scenarios, there is often a point where any further gains are not worth the extra cost introduced into the application. In this case, the

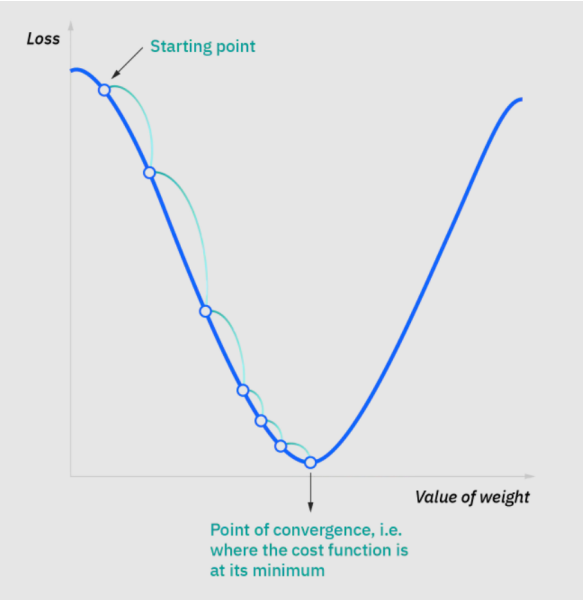

Figure 2.2: Example of Gradient Descent by Education (2020)

point after which the minimum value of this function is reached can be seen, denoted by the point of convergence label.

For image classification, convolutional neural networks are widely considered the best deep learning method; they began gaining in popularity within the last decade, as noted by Z. Chen *et al.* (2018). With a large enough image set, CNNs can automatically extract spatial features from the images using convolutional layers. Each layer has several kernels, these kernels focus on a specific section of the input data to identify higher-level features before moving to the next section to identify high-level features, eventually extracting features for the whole input. These kernels can vary in size, and the stride controls the rate at which they move. The stride dictates how much the kernel moves; for example, if the stride is set to two, the kernel will move two pixels at a time. Once these kernels have scanned the entire input, these values are passed to the next layer; this could be another convolution layer or a pooling layer.

Pooling layers are used on these feature maps to reduce the computational complexity. Each feature map can contain many parameters, all of which are not needed, only the most informative features are kept. The performance of a CNN can be tweaked or enhanced using different pooling methods; the most commonly used pooling tech-

niques are Max or Average pooling, taking either the max or average value per kernel and passing only that value to the next layer. Depending on the size of the kernels used, this dramatically reduces the number of parameters sent to the next layer.

An example of Max pooling can be seen in Figure 2.3; in this example, a 4x4 matrix is reduced to a 2x2 matrix, each 2x2 grid in the original matrix is analysed, and the max value is returned; this generates a new smaller matrix with ideally, the most relevant information from each of the 2x2 grids, denoted in this example by separate colours.

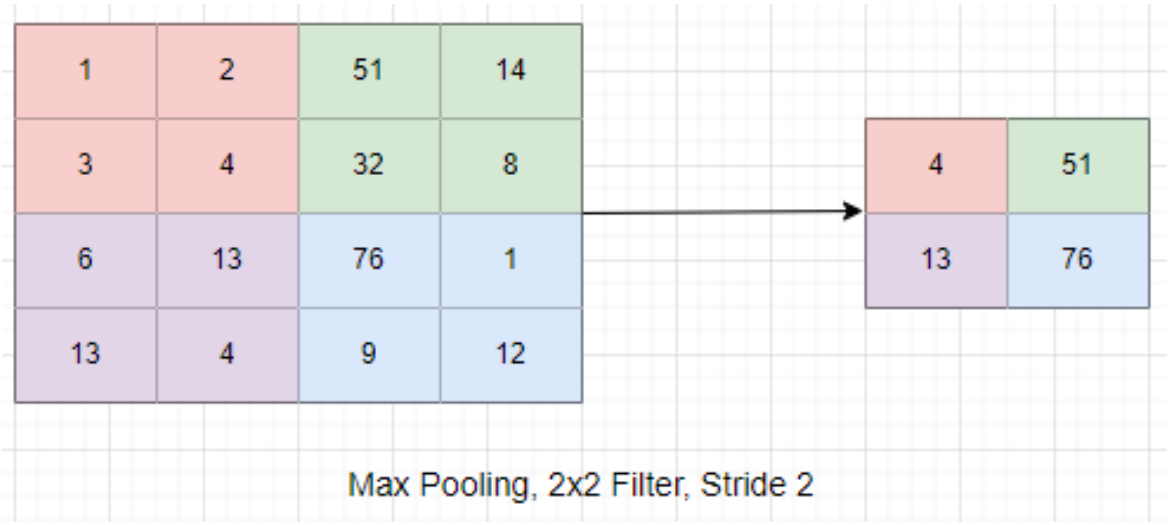

Figure 2.3: example of 2x2 max pooling with a stride of 2

The final step in the image classification process involves one or more fully connected output layers. A fully connected layer means that each neuron connects to every neuron in the previous layer. The main difference between these fully connected layers and a convolutional layer is that the convolutional layers capture local features, such as shapes or lines, whereas the fully connected layers have less flexibility when it comes to learning, these are also called dense layers, due to the dense number of connections. If any of the inputs to the layer change, it can change the outcome. This layer flattens and transforms the feature maps from the previous layers, processed with fully connected output layers to get an image classification output. These final layers are often called a Softmax layer as they only calculate the maximum value to generate a prediction. This output also shows a confidence score for the predicted class labels.

These three layers are the core of a CNN, convolutional layers, pooling layers and output layers; however, there is one other aspect that should also be mentioned: Activation functions. The purpose of an activation function is to decide if a neuron should be active or not. The most common activation function used for CNN based image classification is Rectified Linear Activation Unit (ReLU). This activation function makes it easier for the model to generalise and adapt to unexpected data and most importantly negates all negative value neuron activation's. The ReLU function sets all negative values to 0. These are set to 0 and are not passed to the next layer. This further helps reduce any unwanted parameters being passed between layers and blocks, this helps speed up the gradient descent and reduces the model training time. Another important activation function in CNNs is the softmax function. This function aims to take the output from the fully connected layers and convert this output into a vector of probabilities.

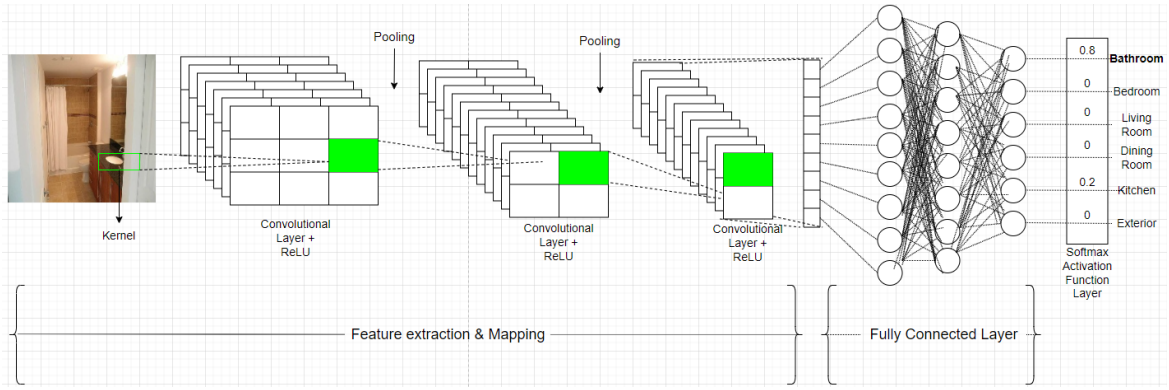

Figure 2.4: Basic Architecture of a CNN

Figure 2.4 shows a idealised example of how a kernel, denoted by a green square, can be reflected through the different layers in a convolutional neural network. In this case, the feature, a bathroom sink, is identified in each convolution layer and is retained through each pooling layer as a relevant feature. This feature, along with others eventually being used in the fully connected layers to help classify the image in question as a bathroom. Note that the probability for the bathroom was 0.8 while the probability for the kitchen was 0.2; the sink feature can be found in both rooms. However, adding other potential features, such as shower enclosure, shower, and shower

curtain, can help correctly classify the image as a bathroom. In reality this sink would not be identified as a sink as the CNN has no knowledge of the object label, but with enough data the CNN would identify the feature as something that is prevalent in the subset of data used to train the specific bathroom class in this instance.

2.1.2 Object recognition

Object detection or recognition is this study's second core computer vision application. Since the two are related, it is better to describe object detection as an image classification technique. Where image classification identifies the image as a whole, object recognition identifies the individual objects within the image, labels them and estimates a location using a bounding box. This estimated location is called object localisation; generally, the object's location can be as important as the classification in an object detection use case. Object recognition can identify a single object from an image or find all available objects from the image. Successful object recognition can be challenging as the objects can be partially obscured or hidden behind the foreground. Object recognition also commonly uses convolutional neural networks, as shown by Pathak *et al.* (2018). For both object detection and image classification, a large enough dataset can help reduce these challenges; with a comprehensive image set, the partially obscured objects become less of a problem. Unfortunately, having access to a comprehensive data set is not always possible, so some image manipulation is needed during training to have a representative sample. Shorten & Khoshgoftaar (2019) explore this need for data augmentation to help reduce overfitting due to a lack of quality training data.

One of the recent developments in object detection has be Region-Based Convolutional Neural Networks, these break and image down into a number of regions, usually 2000 and each region is explored individually, this was quite time consuming to train so Fast-RCNN and Faster-RCNN were developed soon after. Instead of running the CNN on every region, it is now run on the image as a whole first, and then the regions are filtered to identify areas of interest. This meant that Fast-RCNN were significantly faster than RCNN. Faster-RCNN improves on this even further, it uses

the same method of running the CNN on the image as a whole and selection regions of interest, the way these regions are selected is where the improvement is found in Faster-RCNN, this selection process is now delegated to a separate network to predict these regions.

This object detection model is trained on the Open Images V4 dataset and uses a feature detector using Inception ResNet V2. This feature detector is pre-trained on the ImageNet dataset.

- **Open Images V4:** Open Images is a dataset of approximately 9 million images that have been annotated with bounding boxes to show object classification and position. This dataset has been heavily curated manually, with most of the bounding boxes drawn by professionals to ensure high levels of accuracy. The images cover a variety of contexts and, most importantly, contain complex scenes, with an average object count per image of 8.4. As our use cases involve interior household spaces, the ability to extract objects from cluttered spaces is vital.

- **Inception ResNet V2:** Inception ResNet V2 is a CNN similar to VGG16 detailed in Section 3.4.1. However, where VGG16 has 21 total layers, Inception ResNet V2 has 164 layers and can classify objects into 1000 object classes. Inception ResNet is a hybrid model, using the Inception model as a base and augmenting it with some aspects of the ResNet model.

  - The inception model tries to find the best kernel size; a larger kernel size is better at finding globally present features. However, a smaller is better for locally present features. This is the issue that the inception model tries to solve. It performs convolution on every input, with three different sizes of filters 1x1, 3x3, and 5x5.

  - ResNet adds to this again and introduces residual connections, adding the output of the convolution layer to the input. Generally, the input will go through each network layer sequentially in a CNN. The input goes through

a single path with a length equal to the number of layers within the CNN. Residual connections provide an alternate path for data to reach deeper into the CNN by skipping some layers.

2.2 Knowledge Graphs

A knowledge graph is a collection of interlinked entities, concepts, events and relationships. For this paper, The knowledge graph deals with the entities and relationships between those entities. A knowledge graph puts data in context relative to the entire dataset and allows exploration of the relationships between entities. Search engines can use this structured data to help return more accurate results. In the use case, there is also a quantifiable variable that can be extracted from the contents of the room, the frequency of occurrence of an object in a specific room can be used to help identify the room's function.; the number of instances of each object found for each room entity.

Shi *et al.* (2017) details the creation of a textual knowledge graph of medical information. While not in the same domain of study as this paper, it shows the flexibility and power of a knowledge graph. Data is becoming more and more plentiful, and companies are investing more than ever in using that data as a resource; Marr (2021) shows that in 2018, 2.5 Quintilian bytes of data are created daily, with the ever-increasing move toward digital offerings and the global shift to work from home during the Covid-19 pandemic, this figure has likely risen even further.

A basic example of a knowledge graph is shown in Figure 2.5, in this case showing the frequency of objects found in a bathroom. This has been limited to the 15 most commonly found objects for clarity, showing the simple relationship of room to object with the associated frequency count for each link. In this case, it can be seen that the most common objects are the toilet and sink, as well as other common bathroom specific items. Some other items are shown; however, in particular, the window and furniture, these objects are not specific to any one room, so they may cause some confusion going forward.

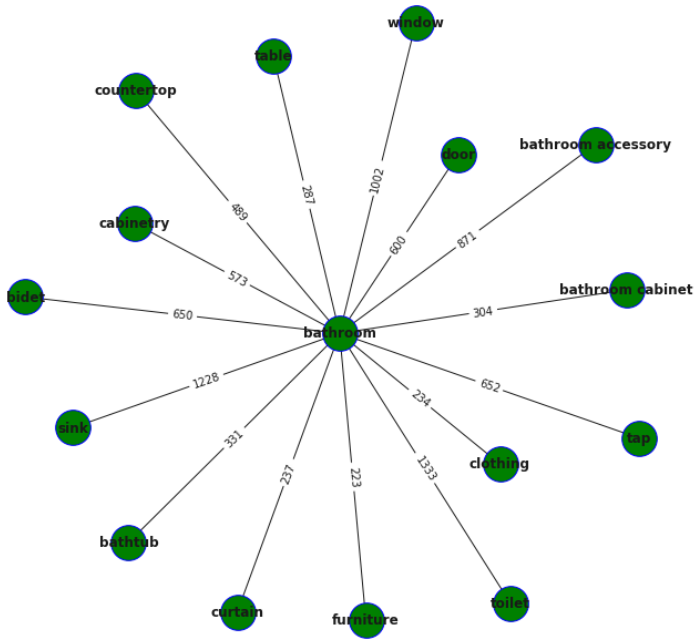

Figure 2.5: Basic Example of a Knowledge Graph

2.3 Developments in Computer Vision

As computer vision techniques have advanced, the traditional convolutional neural networks, detailed in Section 2.1.1 models, have served as a base for more complex implementations.

Roy *et al.* (2020) explore hierarchical classification with incremental learning; generally, a CNN is trained and deployed in a discrete process. If there is a shift in the data or the training is inadequate, it can cause a loss of classification performance. If this were to happen, a new discrete effort would be needed to rectify the problem. Roy *et al.* (2020) proposes adapting a Deep Convolutional Neural Network (DCNN) to allow self-growth. A network of DCNNs would be deployed that could learn and adapt to new information as it is introduced to the system. They describe the network as a tree, growing new branches to facilitate introducing new information while preserving the existing classification performance.

Multi-part Convolutional Neural Networks (MP-CNN) were studied by Divya Meena & Agilandeeswari (2019). This process strived to identify animal breeds from image data; however, it went past simple classification. This method classified breeds at a generic and fine-grained level, for instance, attempting to classify an image as a dog and a greyhound. This approach used a Modified Hellinger Kernel classifier to quantify the similarity between two probability distributions to help identify misclassified data and retrain the MP-CNN to improve accuracy. These developments have achieved an extremely high accuracy rate of 99.95%.

S. Y.-C. Chen *et al.* (2022) recently explored the uses of CNNs in high energy physics. Historically high energy physics has had large amounts of data. Hence, it is an excellent use-case for a deep learning application; They propose leveraging machine learning to provide scalable data analytics. The paper compares a new hybrid quantum convolutional neural network (QCNN) and classical CNNs to see if the new hybrid model can outperform the established methods. They find that the performance is slightly improved between the two methods with the additional benefit that the QCNN takes less time to train to the same level of accuracy as the classical CNN.

2.4 Knowledge Graphs & CNN Integration

The addition of knowledge bases to aid CNN classification, as seen with Wang & Mao (2019) is another recent development in the field. Semantic knowledge bases have been used extensively, most famously by Google with Search. This kind of knowledge base has also seen use in machine learning applications, such as sentiment analysis shown by Kontopoulos *et al.* (2013) and fake news detection applications by Bharadwaj & Shao (2019). However, they have also seen some use within the field of computer vision, both Rubin *et al.* (2009) and Khan (2007) use semantic information to help with image annotation. Image annotation is used extensively in object detection; well-defined annotated objects can improve object detection performance. Using an external semantic knowledge base can highlight the relationships between objects, which can then be used to refine the object detection training data.

Wang & Mao (2019) uses semantic text features to help remove the costly step of curating supplementary data to aid the training process. Any deep learning process is only as good as the data it is trained on, so improvements in the ease of a training process can provide high value. Adding value to the training process is only one use case, Menglong *et al.* (2019) use semantic data and a knowledge graph to aid object recognition, using a semantic refinement method to explore relationships between similar object categories and using this knowledge to build knowledge graphs; these graphs can then be compared using their adjacency matrices to merge classes further.

This paper's primary focus is using an external knowledge base, particularly knowledge graphs, to supplement a CNN to aid the classification process. Dai *et al.* (2021) propose using a graph convolutional network to extract grammatical information, context-specific semantic information, and more general semantic information to help enrich the overall semantics. Adding the more general "common sense" graph information leads to higher accuracy levels and increased efficiency in extracting the structural information of the text. This use case uses two separate knowledge graphs, one for the context-specific semantic information and one for the "common sense" general semantic information. A combination of both can increase the accuracy of two tested datasets.

Meng *et al.* (2019) explore a similar use case, using an existing knowledge graph of data to aid in text classification. Text classification is the fundamental task of natural language processing. This paper also uses a CNN, but it uses a CNN to classify text, unlike the others. As seen in previous examples, this instance uses feature extraction to focus on keywords instead of objects. This process's findings show that adding a knowledge graph helps reduce overfitting and helps improve the accuracy of the model.

Yumeng *et al.* (2021) goes slightly further into the field of image classification, using image classification paired with a knowledge graph to automatically generate a news image caption. The existing methods of caption generation explored by the team were ineffective partly due to the lack of knowledge about the relationships between entities. A knowledge graph is a perfect candidate to fill this gap and proved effective at noticing inconsistencies between news images and their captions. This use-case is

similar to the area of study proposed for this paper, but uses image data and a lexical graph. This study uses also makes use of image data and lexical data but the source of lexical data is unique, this study using object recognition to help build a knowledge graph of this data instead of human intelligence.

2.5 Dataset

With the increase in popularity of hobbyist data science, more and more datasets are becoming available online; kaggle and other data websites have hundreds of datasets available to download with minimal vetting or guarantees of data quality. Issues can arise by using data that is incorrectly labelled or incorrectly annotated. With image data, some datasets are seen as the gold standard, ImageNet, Common Objects in Context (COCO) and the Canadian Institute for Advanced Research (CIFAR). In this paper, ImageNet is used in training the object recognition model and makes uses of the Open Images V4 dataset noted in chapter 2.1.2. The VGG16 convolutional neural network also makes use of the ImageNet dataset, using it to pretrain a model using the ImageNet weights as a foundation.

The ImageNet dataset, introduced by Deng *et al.* (2009) is an extensive image database with over 14 million images as of May 2022. It is organised according to the WordNet hierarchy (an extensive lexical database of English), in which hundreds and thousands of images depict each node of the hierarchy. Currently, only nouns have been mapped, but work is ongoing to increase this further. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a challenge run from 2010 to 2017 to use the ImageNet data to develop object detection and image classification applications. These challenges have produced some of the best performing models for both applications. Recht *et al.* (2019) use the ImageNet database to test if any classifier trained using the ImageNet data can generalise, as there is a risk of over-fitting when the same datasets are used so extensively across the field of study. They found that ImageNet was a good starting point for their development of machine learning applications but was unreliable across different iterations. Even with a simplistic use case, the models

do not generalise well. Indoor Scene Recognition is the primary image dataset used for this study. It is provided by the Massachusetts Institute of Technology (MIT). This dataset is 2.4 gigabytes and contains 15620 images across 67 classes. We use six classes for this study, focusing on the domestic images, not commercial or amenities locations. As this study uses six classes, most of the Indoor Scene Recognition is not being used; the House Rooms Image Dataset is an openly available image set on the data science website Kaggle. It is similar to the Indoor Scene Recognition dataset. However, it is limited to 5 classes, all domestic room images with 5250 images across five classes, all 5 of which are within the scope of this study.

2.6 Gaps in Research

The use of a knowledge base comprised of visual data is still under-explored. As noted in Section 2.3, most knowledge based-systems consist of semantic textual information garnered from natural language processing (NLP) for sentiment analysis or other related text-based applications. Alternatively, an existing semantic knowledge base is used to annotate image data to aid object detection or classification application. Some applications use knowledge graphs to fill a gap in extracted image data, but very few applications attempt to use this data to aid classification. ImageNet and COCO are vast knowledge bases that, if leveraged properly, could provide a fixed reference point to enhance any computer vision application. Suppose an extensive semantic knowledge graph could be developed. In that case, it could act as that reference, being consumed by computer vision applications in the same way the memory acts as a reference for biological vision. Creating a knowledge graph containing raw image data that could be referenced and searched would be the ideal solution. However, the scope is too large for a Master's level project, so a hybrid model that uses image data to generate a text-based knowledge graph is proposed, using object recognition to build a textual knowledge graph from image data.

Chapter 3

Experiment design and methodology

This section will detail the experimental methodology used to determine if the null hypothesis should be accepted or rejected. This study will compare a baseline convolutional neural network with one that has been augmented with a knowledge graph of semantic image data. The same CNN will be used for both the baseline and enhanced model. This is to ensure accurate results by removing variations between each trained CNN. The experimental outcome will clearly show that the addition of the knowledge graph data affects the output of the CNN. All models were trained and tested on a combined indoor scene recognition dataset from MIT and a housing image dataset from Kaggle. The combined dataset consists of 10920 images across six classes, consistent numbers across each class. The ImageNet dataset also plays a key role, being used supplementary in both image classification and object recognition. It is used to train the model this experiment uses for object recognition and the ImageNet weights are used to pretrain the CNN used for image classification Six hundred images are used to generate object-based knowledge graphs for each class using object recognition, 100 images per class. These knowledge graphs can be used as a touchstone to aid uncertain classifications. A naive Bayes model generates the probabilities of an image belonging to each class based on the knowledge graphs generated. The output of this model can then be used to augment the CNN classification.

3.1 Dataset Description

This experiment makes use of 3 main datasets.

1. **ImageNet**: ImageNet is one of the largest image datasets currently available for free for non-commercial use, with over 14.1 million images as of May 2022. This

dataset is used by both the CNN for image classification and the Faster-RCNN for object recognition using ImageNet. The image classifier uses ImageNet to pre-train the neural network before using the final training data. The Object recognition Faster-RCNN has trained on over 1 million images from the ImageNet dataset, with the ability to classify objects into 1000 distinct object classes.

2. **Indoor Scene Recognition**: Indoor Scene Recognition is an image dataset from the Massachusetts Institute of Technology (MIT). This dataset is 2.4 gigabytes and contains 15620 images across 67 classes. This dataset is designed to help generalise classification models, stating, "Most scene recognition models that work well for outdoor scenes perform poorly in the indoor domain". Six classes for this study, focusing on the domestic images, not commercial or amenities locations. This was restricted to only household images, five indoor and one outdoor class, the addition of non domestic pulic spaces would increase the graph complexity and time needed to generate and run the data.

3. **House Rooms Image Dataset**: House Rooms Image Dataset is an openly available image set on the data science website Kaggle. It is similar to the Indoor Scene Recognition dataset but is limited to 5 classes, all domestic room images with 5250 images across five classes. These classes were within this project's scope, so this dataset was used to bolster the number of samples per class. As this dataset is not regulated by an educational institution or some governing body, extra care will be needed in the data cleaning and preparation to ensure the data is appropriate.

3.2 Data Preparation

Data preparation is vital for any data set; thankfully, image data has some well documented best practices when it comes to preparation. Images can have many different conditions when captured, so this variability needs to be added to the training data. Some of the datasets in this study were small enough to validate manually, namely

the object recognition dataset used to generate the knowledge graph. As this dataset is core to the study, the manual effort for this step was needed. Each image present needed to be a representative example of the class. If an image was bordering between classifications, it was excluded. This level of scrutiny was not possible for the other training data due to the volume of images.

For the training set, care was taken to ensure a representative example of the variability in image capture conditions was added.

- **Brightness**: The image's brightness needed to vary. Some images were artificially darkened or brightened to simulate different real-world lighting conditions.
- **Image rotation**: Images are captured at obscure angles or saved with the content flipped. Some samples were randomly rotated up to 90 degrees to simulate this within the training data.
- **Image Width/Height Shift**: The subject in an image is not always centred. Some images were shifted up or down for height or left or right for width by up to 50%.
- **Horizontal/Vertical Flip**: Images can be flipped for various reasons; screen sharing, for instance, often flips the screen. Some images were flipped randomly to account for this mirroring of images.

3.3 Model Selection

This study uses two main models, one for image classification and one for object recognition. Both use a CNN as a base model, as image classification and object recognition are closely related, but both are distinct and use two very different CNN architectures. This section will detail these CNN architectures from a generalised image classification/object recognition view, with added details where necessary.

3.3.1 Image Classification

An established model could be quickly implemented for the CNN image classification. The python based library, Keras, could provide this easily with VGG16.

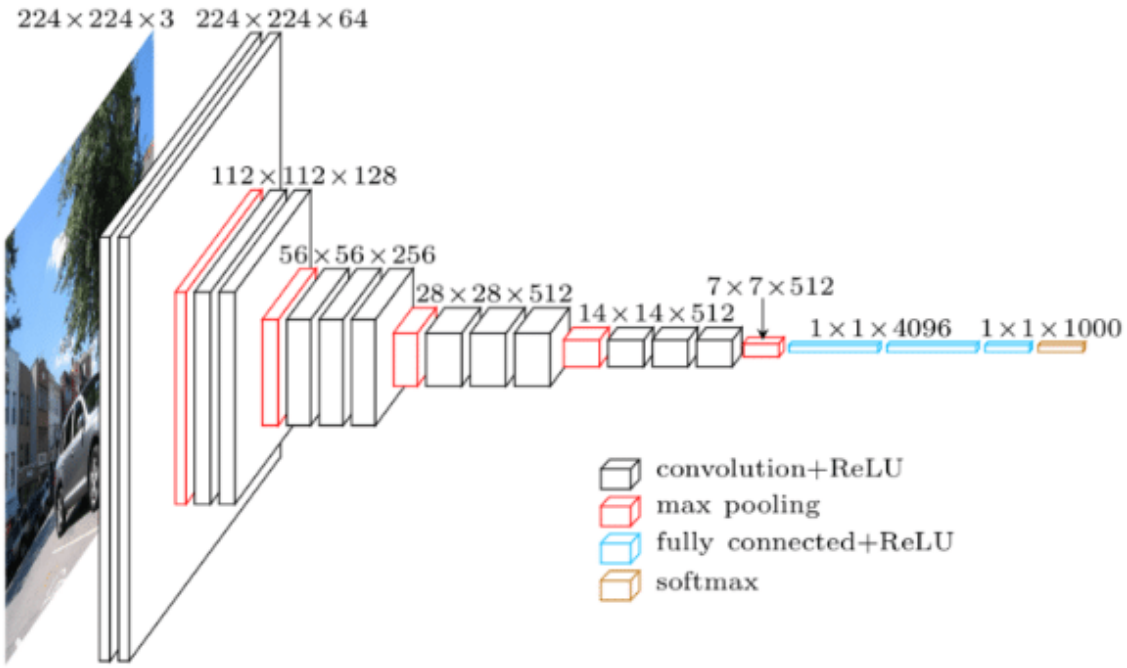

Figure 3.1: VGG16 Architecture Sugata (2017)

VGG16 is a convolutional neural network used to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. It is still considered one of the best computer vision models to date, along with other winners of ILSVRC, such as the 2017 winner SENet and the 2015 winner ResNet. There are 21 layers in the VGG16 model; 16 refers to the number of weighted layers in the model, 13 convolution layers, and three fully connected layers, shown in Figure 3.1 as white and blue layers. However, there are also five non-weighted max-pooling layers, shown in red. VGG16 is structured into blocks, and a max-pooling layer follows each group of convolution layers. As shown in Table 3.1, the only parameters that change between blocks are the number of filters and the resolution.

	Block 1	**Block 2**	**Block 3**	**Block 4**	**Block 5**
Resolution	224x224	112x112	56x56	28x28	14x14
Filters	64	128	256	512	512
Kernel Size	3x3	3x3	3x3	3x3	3x3
Kernel Stride	1	1	1	1	1
Pool Size	2x2	2x2	2x2	2x2	2x2
Pool Stride	2	2	2	2	2
Padding	1	1	1	1	1

Table 3.1: VGG16 Block Parameters

Table 3.1 shows shows a high level overview of the VGG16 CNN block structure, with the relevant parameters for each block, most of these parameters are fixed, with the exception being the number of filters (kernels) and the resolution of each block. Block 1 takes the image at a complete input resolution of 224x224 with 64 filters; each filter creates a feature map to catalogue the presence of detected features. For VGG16, the filter size is 3x3, and the stride is one across every convolution layer, as shown in Table 3.1. As shown in Figure 3.2, with a filter size of 3x3, each filter will analyse a 9 square grid to extract any detected features. These extracted features are then added to the feature map. In each block, the dimensionality of each matrix needs to be consistent, to achieve this padding is added.

Figure 3.2 shows this padding, represented by grey boxes with a 0 value; this padding does not influence the output of feature detection but allows the dimension of the matrix to be maintained between convolution layers; in this case, the red 3x3 grid maps to the single green element in the feature map. When the 64 filters in block one have generated their feature maps, a max-pooling layer reduces these maps to reduce complexity and the number of parameters passed forward. The max-pooling layers in the VGG16 use a 2x2 filter with a stride of 2. Since this padding is not present on the Max-pooling layer, it demonstrate how a max-pooling layer can reduce a 4x4 matrix down to a 2x2 matrix using the max value from every 2x2 grid. This reduces

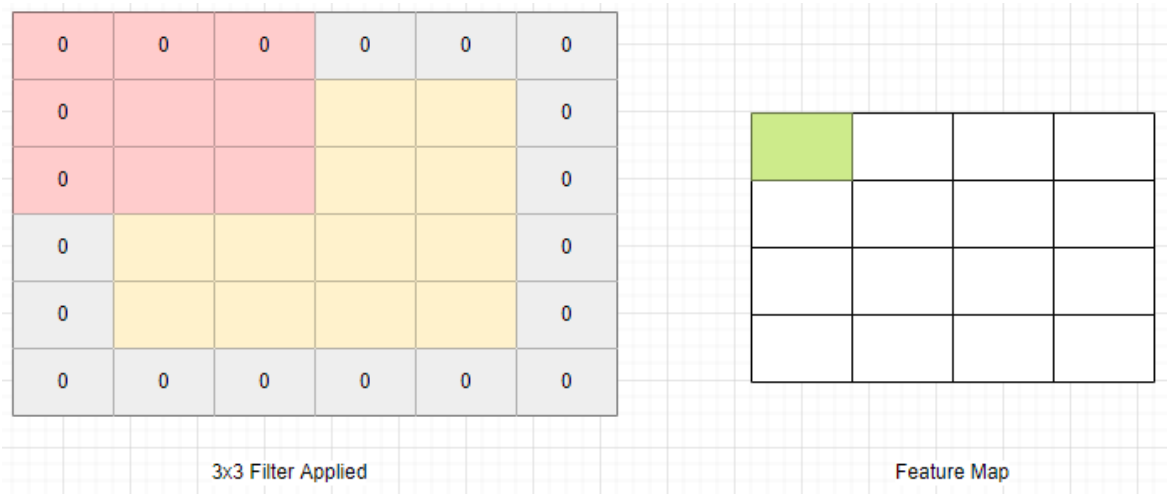

Figure 3.2: Feature Extraction with a single 3x3 Filter

the resolution of the input by half, down to 112x112. This is then passed into block 2.

Blocks 2 to 5 are functionally the same as block 1. They consist of several convolution layers with a max-pooling layer to reduce the number of parameters passed out of the block. Each block reduces the image's resolution but increases the number of filters. Increasing from 64 filters in the first block to a max of 512 filters for blocks 4 and 5. This increase in filters allows the CNN to extract more features in parallel from the input. As these filters increase, the resolution decreases. It may seem counterproductive to reduce the resolution of image data that is being classified or trying to extract information. However, this reduction is more than just deleting half of the information available. The max-pooling operation should extract the most meaningful data from every 2x2 filter. The hope is that as the blocks progress, the quantity of meaningful information will be higher than in the preceding blocks. The ReLU activation function is used on every convolution layer across all blocks.

After the five blocks of convolution and pooling, there are three fully connected layers. It must be flattened before the output from the final max-pooling layer can be used in these fully connected layers. This converts the data to a one-dimensional array, a single feature vector that can be connected to the fully connected layers. VGG16 uses three fully connected layers, two of which have 4096 neurons each, with the final

layer containing the number of classes trained on our classifier. For the ImageNet example, this final fully connected layer has 1000, but in the case of this study, 6. The last step in the classification process is using the softmax activation function.

3.3.2 Object Recognition

As with the image classification model, an established model for object detection was needed. There are multiple available online, although not all fit the use case of this study. The model chosen is the Faster Region-Based Convolutional Neural Network (Faster-RCNN) developed by Google. This object detection model is trained on the Open Images V4 dataset and uses a feature detector using Inception ResNet V2. As this object detection model is trained on a comprehensive dataset and the feature detector uses a model with good performance, the risk of the model being trained on an unrepresentative dataset as shown by Li (2021) is mitigated, so is a good fit for this study.

3.4 Training & Testing

The development for this study was performed using Google Colab and the python programming language. This study is computationally heavy; the Google Colab Pro tier provided larger RAM and faster GPU access. This helped reduce the time needed for training. Google Colab also helped with file management, and Google drive allowed the images to be stored in the cloud. The image data was split into three distinct categories. Object Recognition, Training, and Testing were further divided into each class for testing. Bathroom, Bedroom, Dining Room, Exterior, Kitchen and Living Room. As the images were sourced from multiple datasets, the image size was inconsistent across the samples. This size disparity is addressed in the image preprocessing.

Data Preprocessing

The Keras library is used for it's preprocessing functionality needed for the data augmentation, as noted in Section 3.3. This function allows the data preparation to be

performed as the image data is loaded into a training set and reduces the need for actions on an image by image basis. A similar function is used for the testing data, but this excludes the data preparation steps outlined in Section 3.3. These preparation steps are performed on the training data only, to help the model generalise and expose it to the greatest variability in the samples provided as possible. As data representative of real-world conditions is needed for testing, the preparation step was excluded for the testing data, if this data went through the same perpetration steps as the training data, it may hurt performance by creating a model that performs well when testing on this data, but does not match that performance in a real-world application.

The ImageDataGenerator used helps create a Directory iterator to set some of the VGG parameters as the training, testing and validation sets are created. The image size, 224x224, is set here, which resolves the issue of varying image sizes across datasets. This accomplishes this by using bi-linear interpolation to resize the images down or up to a standard size, this uses the existing pixel values to determine the missing values when scaling up an image and determine the updated pixel values when scaling an image down. Batch size is also set at this stage, in this case, 64; this represents the number of images used to train the model at any time, this value was chosen as it has been found to work well, as noted in a study by Google Smith *et al.* (2018). A random seed is set for each run, which will be used for a Monte Carlo simulation. This random seed ensures that the same training/testing split is not used across all training and testing cycles. A separate Directory iterator is used for the validation and testing data. With the datasets now in place and ready to be used, the training and testing process can be broken down into distinct steps.

- Use training Data to fit the VGG16 model for image classification.
- Use the Object Recognition data set to create knowledge graphs for each classification class.
- Create a Naive Bayes model based on knowledge graph data.
- Combine all aspects to aid VGG16 classification.

VGG16 Implementation

The python programming language was used to implement the VGG16 model for image classification, using the Keras and TensorFlow libraries. The number of epochs the data was trained with for model training needed to be decided. An epoch is a hyperparameter that controls the number of times the algorithm will run through the entire training dataset. The number of epochs can significantly affect the accuracy of an image classifier, but there is usually an epoch that is a good stopping point. Both the accuracy and loss will be used to make this decision. Using the graphs in Figure 3.3 can help visually show this learning curve. Focusing on the validation line gradually improves as the number of epochs increases, but 19 is a good stopping point. At this point, the accuracy level is high, and the loss is low, the VGG16 model uses the categorical cross-entropy loss function as this is commonly used for multi-class classification problems. After this epoch, the accuracy and loss still improve, but both have added instability. This epoch number is also relatively low so that each training cycle can be completed in a reasonable time.

Learning rate is another hyperparameter vital for deep neural network training. This sets the rate at which the model adjusts it's weights in response to errors. If the learning rate is too low, it can result in a lengthy training process, but if the value is too large, it can result in sub-optimal performance, inconsistent training results and a failure to converge. Thankfully some guidelines can be followed. TensorFlow recommends a learning rate of 0.001. This value was chosen because gaining the peak performance out of VGG16 is not the fundamental goal of this study. There will be a Monte Carlo aspect to this training and testing cycle; the computational time needs to be achievable in a limited time frame. Both the number of epochs and the learning rate affect this. The goal is to find the middle ground where good performance is achieved with minimal resources.

Once this training is done, the model can be used to classify the test set. This produces an array of classification labels. This array is compared to our true labels for the test set. This allows an accuracy score to be calculated for the prediction. This

classification score is the baseline for each trained model.

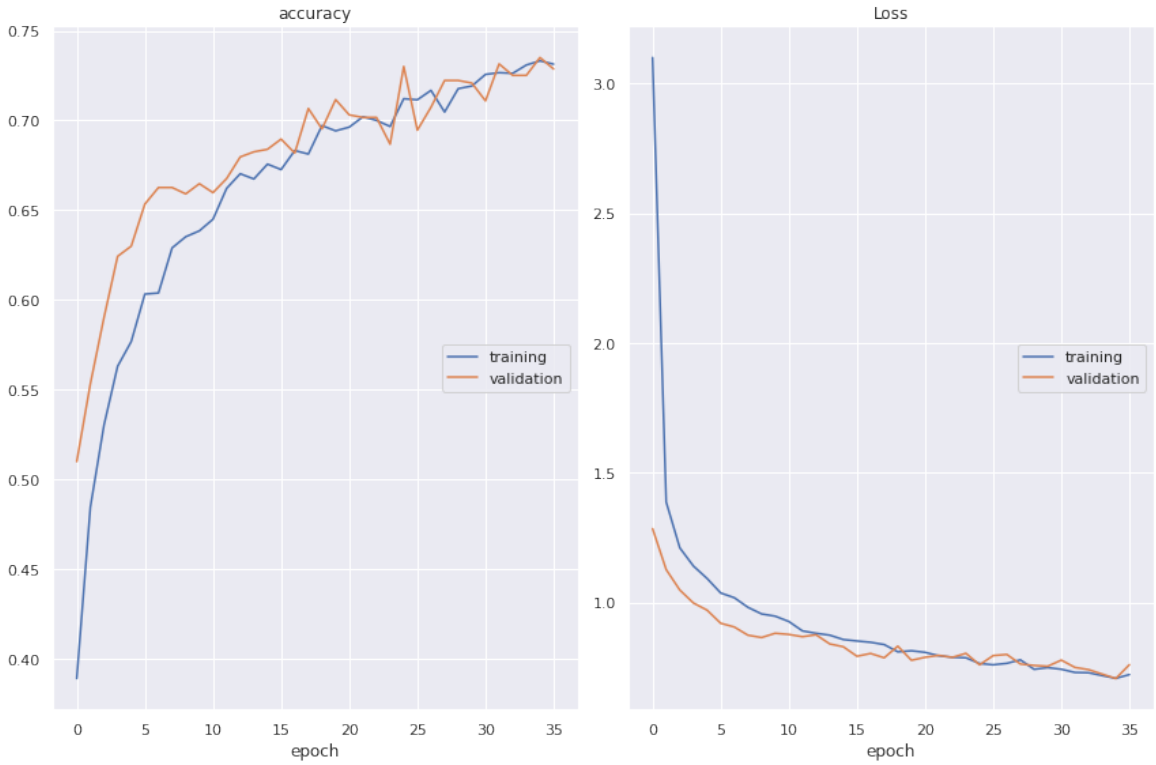

Figure 3.3: Number of Epochs run for VGG16

Object Recognition Implementation for Knowledge Graph

With the pre-trained object recognition model, no training process is needed. The next step is integrating this object recognition model into generating a knowledge graph. The object recognition dataset consists of a fixed number of images per class to remove any levels of frequency bias as the knowledge graph will record the number of occurrences of a specific item per room. The object detection model was used to process every image in the object recognition dataset and record the findings in a data frame. This data frame recorded the room the object was found in, which can be retrieved from the file structure naming, and the object found. This results in a two-column data frame that needs some processing to be usable in a knowledge graph. There will be multiple entries for any given room and object combination; this must be consolidated and counted. A group by function allows this to be performed easily. This

is a deterministic process; this data frame was also saved to a CSV file to eliminate the need to run this process for every training pass. With this data, knowledge graphs can be created using the networkx python library; this can take a data frame to generate a graph and add nodes and edges to link these objects to rooms. The frequency can also be used here to label each of the edges.

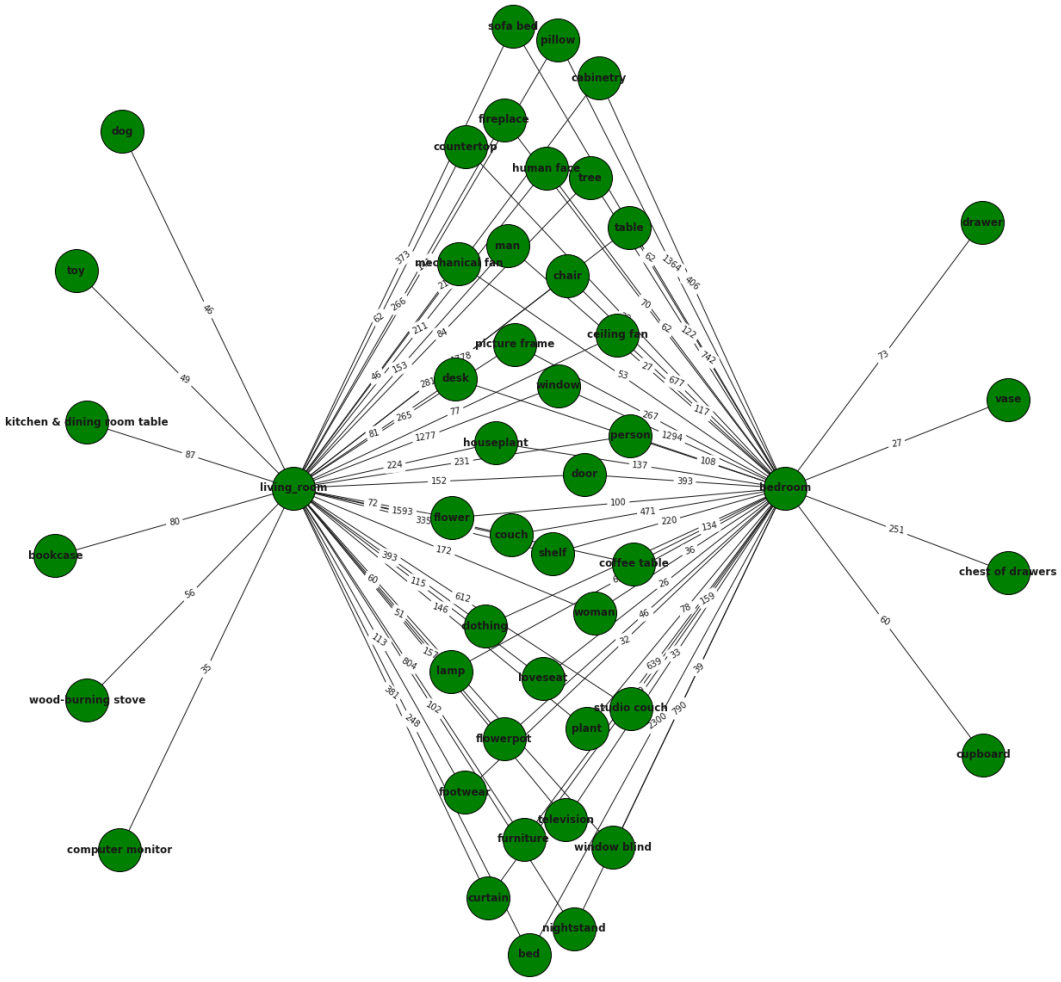

Figure 3.4: Knowledge Graph showing both Living Room and Bedroom object frequency

A knowledge graph showing both the living room and bedroom objects can be seen in Figure 3.4. This shows the individual objects for both the living room and bedroom

image data. It shows that the majority of objects are found in both rooms, with the only objects being unique to each room all have a relatively low frequency compared to some of the common items; in particular, furniture, window and pillow stand out as some of the most frequent objects that are found in both the living room and bedroom. This makes sense when applied to a real-world scenario, as most houses' living rooms and bedrooms are very similar in furnishings and decoration.

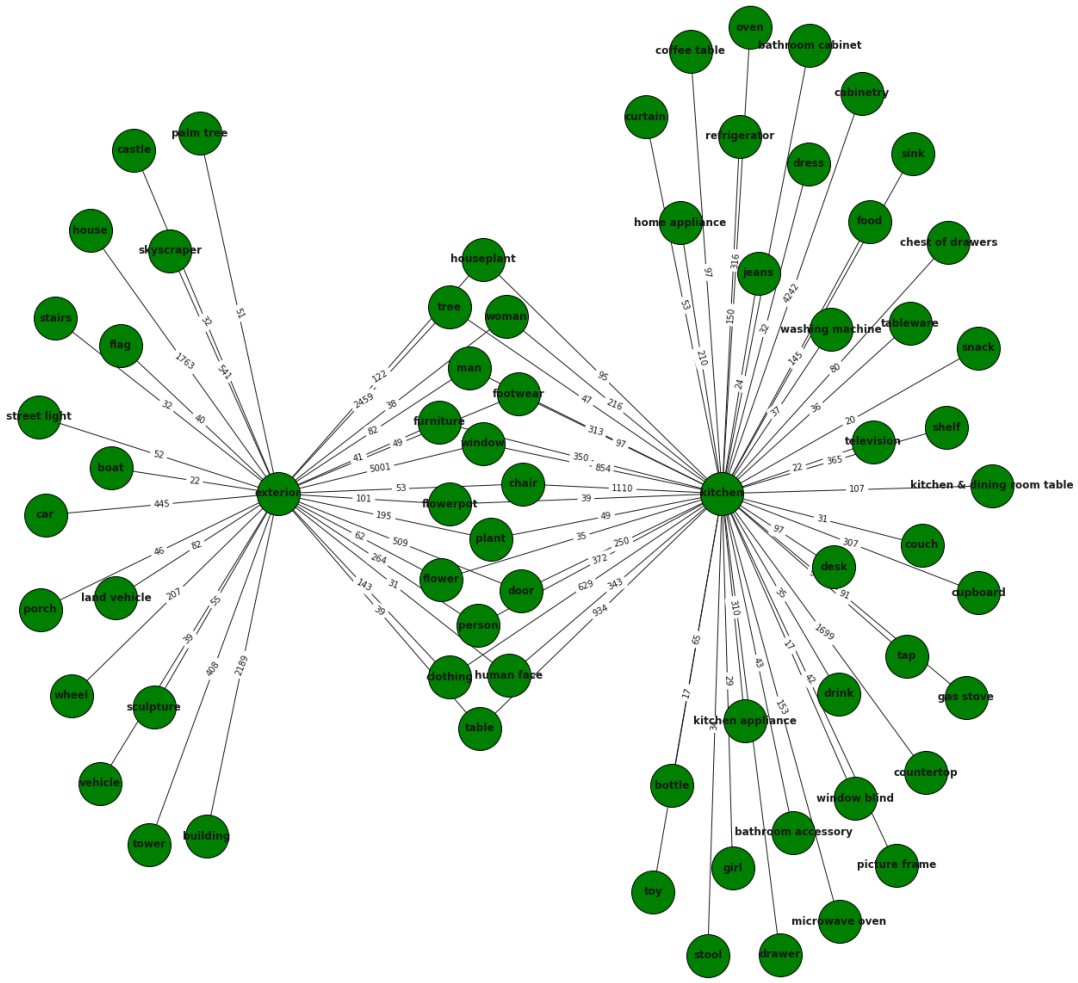

Figure 3.5: Knowledge Graph showing both Kitchen and Exterior object frequency

In contrast to Figure 3.4, the knowledge graph in Figure 3.5 shows a more definitive distinction between the objects found in each of the classes image data. In this

instance, a much smaller subset of objects are common between both classes. There is some overlap, but this frequency tends to skew to one of the classes, for instance, the tree object. This object is found far more commonly in the exterior class than in the kitchen class. The knowledge graphs in the above Figures 3.4 and 3.5 show the stark differences in relationships between the two classes. With the addition of each extra classes, the relationships and overlap increase in complexity. This complexity can also increase as other edges are added; in this case, each object node can have a maximum number of 6 edges, assuming the object is found in every class. If another type of edge is added, for instance, linking objects within the same family, this can add a huge number of links between the nodes. As this complexity increases, the human readability of the graph diminishes, but the level of insight garnered from the graph can increase.

Naive Bayes Implementation

Some means of using the information available within the knowledge graph were needed; A naive Bayes classifier is well suited to this task. Using Naive Bayes, creates an ensamble and allows the knowledge graph data to be used probabilistically. Using the information within the knowledge graph, the Naive Bayes can be trained to recognise the different object distributions from room to room. This would allow the output from the classification model to be checked against the Naive Bayes model. If it was found that both the VGG16 classification was low enough confidence and the Naive Bayes probability was high enough, this classification value could be changed.

This means that the naive Bayes needed to be trained on the entire knowledge graph, not just a single room. With this, the naive Bayes should be able to take an array of items and provide an array of probabilities for classification.

Combining VGG16 with a Knowledge graph using Naive Bayes

In order to successfully join the VGG16 CNN with the knowledge graph, a threshold needs to be determined. The CNN is still the primary classification tool. It is not being superseded by the knowledge graph and the trained naive Bayes model, and this is

intended to complement the CNN, not override it. Some testing is needed to determine this threshold, but firstly the role of the threshold must be defined. This threshold will determine if an image that the CNN has classified should be double-checked using the knowledge graph and naive Bayes; the higher the threshold, the more items will use this double-checking procedure to adjust the classification label. For example, a threshold of 70% would mean that any image classified with classification confidence below 70% would use the knowledge graph and naive Bayes classifier.

To determine the best threshold to use, the classification process was run a total of 75 times. A CNN was generated using randomly seeded testing and validation data from the dataset; this was then used in conjunction with the knowledge graph implementation at various thresholds, from 0.35 to 0.7; this range was selected as initial experimentation. Anything below the lower range showed no difference in accuracy, and anything above the upper range negatively affected accuracy overall. This range was sufficient to highlight the experimental process and determine the desired threshold.

Confidence Interval	**Avg Baseline Accuracy**	**Avg Enhanced Accuracy**
0.35	81.54%	81.57%
0.40	81.54%	81.59%
0.45	**81.54%**	**81.61%**
0.50	**81.54%**	**81.61%**
0.55	81.54%	81.60%
0.60	81.54%	81.60%
0.65	81.54%	81.60%
0.70	81.54%	81.58%

Table 3.2: Classification Confidence Threshold Testing Results

As shown in Table 3.2, the difference in average accuracy is very slight overall, with all of the thresholds showing a slight increase over baseline performance. In initial testing, thresholds above 70% were explored, each showing consistent drops in

accuracy performances as the threshold increase. This indicates that the knowledge graph performs better on images where the CNN is less confident. In this case the best threshold is somewhere between 0.45 and 0.50.

The next step involves using the naive Bayes classifier; as this classifier produces a probability score for each item passed in, a minimum score is needed. This score will try to minimise the misclassifications introduced at this step. This process is similar to the process used to generate the results in Table 3.2, taking a probability score threshold ranging from 0.35 to 0.70 and testing to see which generates the best results. This score is taken by examining the detected items in an image and calculating the probability of appearing in a given room. An average probability is generated. This value is used to determine if the classification label should be changed.

Probability Score	**Avg Baseline Accuracy**	**Avg Enhanced Accuracy**
0.35	81.16%	80.74%
0.40	81.16%	80.98%
0.45	81.16%	81.16%
0.50	**81.16%**	**81.25%**
0.55	**81.16%**	**81.25%**
0.60	81.16%	81.20%
0.65	81.16%	81.16%
0.70	81.16%	81.16%

Table 3.3: Naive Bayes Probability Score Testing Results

Table 3.3 shows the results of this testing, again there is only a very small change in accuracy overall, with a range between 0.5 and 0.55 providing the best results. If the probability score is too low, a larger number of image classifications are adjusted, which causes the accuracy to drop, on the other end of the scale, if the probability score is too high, very few of the image classifications are adjusted and the accuracy remains comparable to baseline. With all of these processes in place, the final step is to fully implement the solution to provide both baseline and enhanced accuracy. Both

use the same CNN, which can be done in one process chain, baseline generation and testing, enhanced generation and testing and then analysis. A Monte Carlo simulation is used to generate a distribution of accuracies for both models. The process follows the following steps.

1. VGG16 is trained on training data using a random seed.
2. VGG16 predicts the classification of the test set.
3. Classification results are recorded for baseline model performance.
4. Knowledge graphs are generated using the stored Object recognition data CSV files.
5. Naive Bayes model is trained using a knowledge graph of object frequency data.
6. Classification results are analysed to identify any with low confidence, a threshold of 50% as noted above in Table 3.2.
7. Identified low confidence results are passed through the Object Recognition model.
8. Detected objects are passed into the Naive Bayes model to generate an array of probabilities.
9. Confidence and Naive Bayes results analysed.
10. Classification value replaced if probability above a certain threshold, 50%, as noted above in Table 3.3.
11. Enhanced Classification Results are recorded for Enhanced Model performance.
12. Both baseline and enhanced performance accuracy were recorded for analysis.

Each iteration within the Monte Carlo simulation uses a different subset of training data, as 15% of the training data is held as validation data during training. The seed provides randomness to the data augmentation processes during data augmentation.

These steps help increase the range of outcomes and help increase confidence in the overall results.

3.5 Summary

To conduct this study, specific vital components must be in place. The targeted datasets must be verified and augmented to represent training examples. The VGG16 model, provided by Keras, is trained on this labelled image data, consisting of various room images and a baseline accuracy generated. The faster RCNN developed by Google, the object recognition model is used to analyse the relevant subset of images and generate a knowledge graph with this information. This graph shows the relationships between room and item and the frequency of occurrence. This graph can then train a Naive Bayes algorithm to provide insight into the object distributions between rooms. Finally, all of these come together to augment the classification output from the VGG16 model. This results in two accuracy scores, one baseline and one enhanced. This process is repeated to generate a good range of samples to increase confidence in the findings.

Chapter 4

Results, evaluation and discussion

The experiment's outcome will be examined and used for hypothesis testing in this section. As a Monte Carlo simulation has been performed, a distribution of experiment outcomes has been generated for the baseline and enhanced classifiers. The accuracy is calculated by comparing the predicted classification against the true classifications for each class; if the true and predicted class match, it is a true prediction. A t-test will be used on these accuracy results, and each model will be explored to see if any classes outperformed the others in either scenario. The hypothesis posits an increase in accuracy, but both an increase and decrease will be explored to fully gauge the performance of adding a knowledge graph to the VGG16 CNN.

4.1 Results

	Number of Samples	**Average Accuracy**
VGG16 Baseline	576	80.88%
VGG16 Enhanced	576	80.98%

Table 4.1: VGG16 Monte Carlo Results with full KG

The test data used to generate these results consist of 1200 image files, evenly split across the six classes, with 200 images per class. This data is used for both the baseline and enhanced models as the enhanced model is a derivative of the baseline. None of the hyperparameters of the CNN was altered between the separate runs. These results, shown in table 4.1, show a minor accuracy increase over a large number of samples, but a single-tailed t-test is needed to find if this increase is significant. The enhanced

VGG16 CNN that received information from a knowledge graph (M = 0.8098, SD = 0.0173) compared to the baseline VGG16 CNN (M = 0.8088, SD = 0.0351) showed no significantly better classification performance, t(1152) = -0.5778, p = 0.2818. As the P-value for this t-test is greater than the alpha = 0.05, the null hypothesis cannot be rejected in this instance.

4.2 Evaluation & Discussion

4.2.1 Statistical Measures

Even though this study could not reject the null hypothesis, adding a knowledge graph did not cause a noticeable performance decrease. The baseline and enhanced performance can be examined to determine if any patterns have emerged from testing. These results are generated using the same methods like the Monte Carlo simulation but are not part of the results set. Figure 4.1 shows the confusion matrix of the baseline

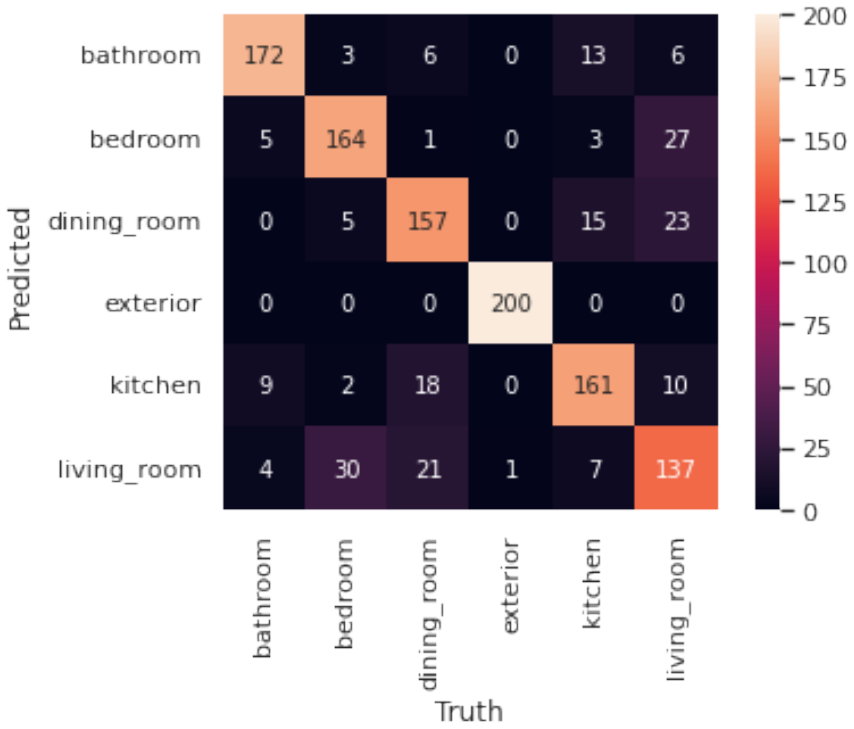

Figure 4.1: Confusion matrix of Baseline VGG16 CNN

VGG16 CNN model; this visually shows the number of correct and incorrect classifications broken down to each class. The two most commonly confused classifications are the living room and bedroom, and the best performance is the exterior class, with 100% accuracy. Living rooms and bedrooms are often decorated similarly, so a certain level of confusion is expected. At the same time, the exterior class is visually distinct in the set; it is the only one based outdoors and has very distinctive characteristics. Figure 4.2 shows the confusion matrix for the enhanced model that partners with the

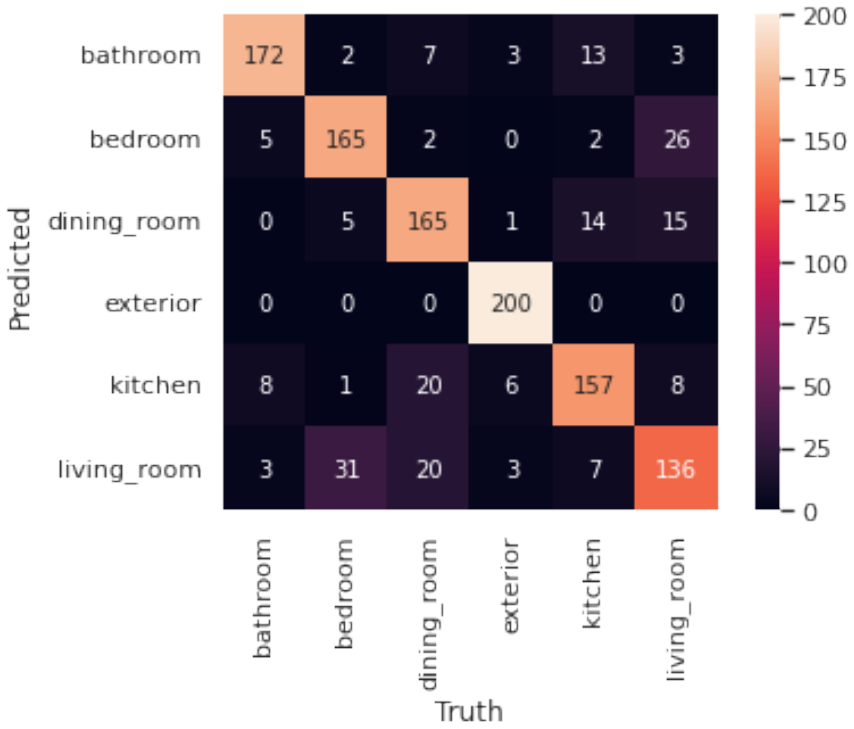

Figure 4.2: Confusion matrix of full KG Enhanced VGG16 CNN

model in Figure 4.1. In this case, the enhanced model has slightly better performance than the baseline, with an increase in the correct classification for the bedroom and dining room classes, the same performance for bathroom and exterior classes but a drop in both the living room and kitchen classes.

These results may indicate that the issue in classification in both the baseline and enhanced models is items found in more than one room, like the bedroom and living room. The enhanced model did not help with the levels of miscalculation between these classes. The bedroom and living room classes show the most confusion in this

instance. The performance of the exterior class is consistent between both models, as this class is a distinct class within the dataset. This can be further explored to investigate if isolating the unique items per room can increase accuracy overall.

To test this, whether this isolation does lead to an increase, a way to refine the knowledge graph is needed. Each room has a list of items and their frequency; this can be used to limit the items in each knowledge graph to some degree, in this case, to the top 25% of items found in each room. This is intended to remove some of the confusion between rooms and lead to increased accuracy. This focuses the knowledge graph on only the items with a greater frequency of being found. As there are varying items found for each room, a restriction based on the count would generate inconsistent results; limiting the items to greater than 50 occurrences for example, could have drastically different effects for each class, making it percentile based resolved this issue and ensured there was still a distinct selection of items for each class.

	Number of Samples	**Average Accuracy**
VGG16 Baseline	119	80.98%
VGG16 Enhanced	119	80.92%

Table 4.2: VGG16 Monte Carlo Results with 25% Limited KG

Running a t-test on this data, shown in Table 4.2, show a very minor accuracy decrease. The enhanced VGG16 CNN that received information from the 25% limited knowledge graph (M = 0.8092, SD = 0.0083) compared to the baseline VGG16 CNN (M = 0.8098, SD = 0.0088) showed no significantly better classification performance, t(236) = 0.4957, p = 0.3103. As the P-value for this t-test is greater than the alpha = 0.05, the null hypothesis cannot be rejected in this instance either.

Looking at the confusion matrix generated using the limited knowledge graph, shown in Figure 4.4, there is a minimal improvement compared to the confusion matrix shown in Figure 4.2. The results are roughly consistent between both samples, showing that the most significant confusion still exists between the bedroom and living room classes, there does seem to be a small decrease in misclassifications overall but this

would need further investigation to determine its significance.

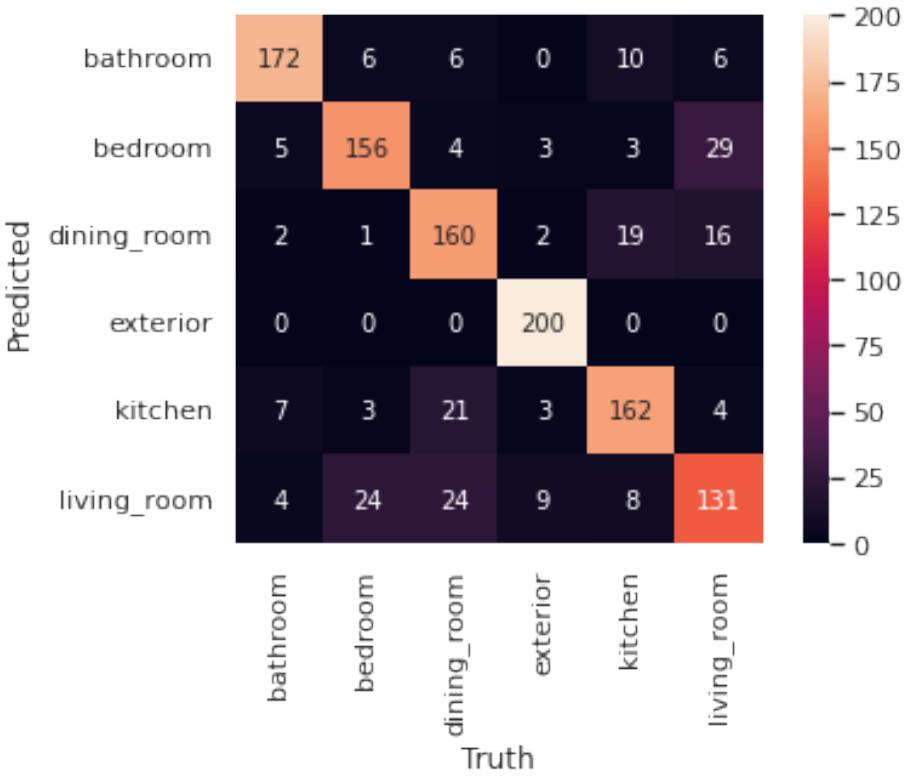

Figure 4.3: Confusion matrix of 25% limited KG Enhanced VGG16 CNN

4.2.2 Image Analysis

Even without a statistically significant performance increase, the model does have some value. Figure 4.4(a) shows an image of a living room. When analysed by the VGG16 CNN, this image was classified as a dining room. It does have many elements that a standard dining room would have: chairs, a table, plants outside, and windows, but from the image, it is clearly a room focused around the small coffee table in the centre. All the sofas and decorative items in the shot frame this coffee table. The knowledge graph trained naive Bayes was able to determine that while a lot of the elements in this image are found in a dining room, the unique elements found point to its actual function, a living room. Most common dining rooms have the dining table central to the room, unlike the living room image with the sofas and coffee table as

(a) Living Room Image example

(b) Kitchen Image example

Figure 4.4: Image analysis examples.

the centrepieces.

To counter the image in Figure 4.4(a), Figure 4.4(b) shows the opposite situation. In this case, the VGG16 CNN correctly classified the image as a kitchen, but the KG trained naive Bayes misclassified the image. The object detection finds chairs, a table, furniture, cabinets and a home appliance. This leads to the updating of the classification of the dining room. Except for the home appliance, which in this case looks to be an inbuilt dishwasher or washing machine, the rest of these items are commonly found in a dining room. This may point to a flaw in the system that could be further explored. If the items had weighted values per room, similar to the way TF-IDF (term frequency-inverse document frequency), this looks at the number of times a word is found in a document and the number of times the same word is found across all documents, this can lend greater weightings to unique words. A similar solution may be applied to this classification problem, finding a single item like a refrigerator, for example, may be more significant than finding six chairs and a table.

Chapter 5

Conclusion

This chapter notes the conclusions garnered from the study. It provides a high-level review of the research overview, the experimental design and methodologies, and an evaluation of the results. It also analyses the influences and contributions of this study and any recommended future work that can be undertaken using this study as a foundation,

5.1 Research Overview

The main objective of this study is to investigate if adding a knowledge graph of image-based semantic data can aid in image classification; this can be divided into four sections. Data gathering and preparation, Conventional Neural Network generation, Knowledge graph generation and finally, the addition of the knowledge graph to the CNN to aid in classification. The datasets used for this study are the House Rooms Image Dataset and the Indoor Scene Recognition dataset. This data is used to train a VGG16 convolutional neural network, and used to generate a knowledge graph of object data; this knowledge graph is used to determine if the classification can be improved using Naive Bayes to try and increase overall image classification accuracy.

5.2 Problem Definition

Many semantic knowledge bases are used to aid classification problems, as noted in Section 2. In particular natural language processing makes use of this. However, the use of this type of knowledge base is under-explored in aiding image classification. This study explores this under-explored area to see if a knowledge graph based on

image-based semantic data can aid in overall image classification by a CNN. though not statistically significant, the results suggest that with sufficient refinement this type of knowledge graph could aid image classification as a whole, similar to how some of the larger image datasets are used for object recognition, like the ImageNet dataset.

5.3 Design/Experimentation, Evaluation & Results

The datasets House Rooms Image Dataset and the Indoor Scene Recognition dataset, were combined to generate an adequately sized dataset for this study's purposes. This joined dataset was subdivided into sections for CNN training and testing and for Object recognition to aid the creation of a knowledge graph. This dataset consisted of 12249 image files split across six classes. Each class was balanced to ensure that the training, testing and object recognition data were consistent.

The VGG16 architecture was used to create a CNN and trained using the created dataset; this model was used for both the baseline and enhanced results. This model was trained and tested to get a baseline accuracy for complete classification. Object detection was used to generate a knowledge graph of image-based semantic data. This knowledge graph was then to train a Naive Bayes classifier.

With all these parts in place, the CNN, the knowledge graph, and the Naive Bayes classifier were then used to determine if the addition of the knowledge graph to the CNN can improve classification accuracy. A Monte Carlo simulation was run to generate an accuracy distribution for both the baseline and enhanced model; While a t-test was unable to discern a statistically significant difference in performance, the enhance model slightly outperformed the baseline, suggesting that with further refinement, this could be a viable technique for improving model predictions.

5.4 Contributions and impact

The reliance on computer vision will only increase going forward. As requirements increase for live image classification, adding a knowledge base to reference can be useful.

This study shows that a knowledge graph of image data can be used in conjunction with a CNN to augment image classification results. Although not shown to be statistically significant, the enhanced model displayed a slightly higher accuracy than the baseline. This shows promise for further improvements in future. The innovation of the work is the generation of this knowledge graph and in using this knowledge graph as a key component in an ensemlbe of models; knowledge graphs have been used extensively for machine learning applications but are lacking in the domain of image classification. This work has the potential to act as a foundation for applications that use both a CNN and a knowledge graph to aid in image classification.

5.5 Future Work & recommendations

For future work on this topic, the main innovations should come around the knowledge graph and how it integrates with the CNN. The knowledge graph in this study is simple, mapping an object's relationship to the room in which it was found. This could also be increased to introduce a relationship between objects; if some objects are commonly found together in certain rooms, finding one on its own could provide significance to aid in classification. These semantic object relations could also help provide a higher-level analysis; many objects are finely classified. However, if a more generic hierarchical knowledge graph was developed, it could help provide initial insights into the room function.

The accuracy shows for the Exterior class, 100% across multiple tests, can be attributed to the distinctive nature of the class within the dataset. This can be further explored to investigate if reducing the objects defined in the knowledge graph to more distinct objects can help improve accuracy. This reduction was explored in this study but was made semi-random, reducing the objects defined in each room based on frequency. This can be adapted to a more deterministic solution by viewing the dataset as a whole to isolate the core items in each room; for example, a bidet is very unlikely to be found in any room outside of a bathroom. This could be a good application for a TF-IDF approach, instead of evaluating how relevant a word is in a

set of documents, it could be applied to an object in a set of images.

Another potential improvement to this would be how the CNN integrates with the knowledge graph; the addition of the knowledge graph, in this case, is done through a Naive Bayes classifier; this adds another variable to the solution that may be altering the classification accuracy. This step was introduced due to this study's time limitations. A more integrated solution to this could help identify the exact impact of the knowledge graph and could help grow the solution further.

As noted in Section 4.2.2, a different way to improve this solution could come with adding a weighted value for each object to determine the critical objects for each class; however, this would add some manual work to determine these weights. If a single toilet is found in a room containing items commonly found in a bedroom, should this toilet shift the probabilities from the bedroom to the bathroom? This would involve isolating items iconic to each class and determining the overlap with other rooms. Generic items such as furniture and windows would provide little insight into the room's function, so these could potentially be removed to focus the model on items commonly used to define the room—for example, toilet for bathroom, refrigerator for kitchen, and bed for bedroom. Careful consideration of the selection process for these critical items would be needed. For instance, some households have toilets and showers in garages or outdoors. This could lead to more granular classes; a room with just a toilet and sink could be classified as a toilet, but with the addition of a shower or bathtub, this classification becomes a bathroom.

One of the major drawbacks of a neural network is the black box nature of their results, in a world where everyone is becoming more data conscious and stricter restrictions are in place, such as GDPR, the addition of a knowledge graph to this can help model the data directly in a way that is easily understandable to humans, this can potentially helps us leverage more deep learning methods without sacrificing the ability to explain why our model arrives at a certain classification or decision.

References

Advances in Computer Vision. (2020). Cham: Springer International Publishing. doi: 10.1007/978-3-030-17795-9

Bharadwaj, P., & Shao, Z. (2019). Fake News Detection with Semantic Features and Text Mining. *International Journal on Natural Language Computing*, *8*(3), 17–22. doi: 10.5121/ijnlc.2019.8302

Brown, J., Oldenburg, I. A., Telian, G. I., Griffin, S., Voges, M., Jain, V., & Adesnik, H. (2021). Spatial integration during active tactile sensation drives orientation perception. *Neuron*, *109*(10), 1707–1720.e7. doi: 10.1016/j.neuron.2021.03.020

Chen, S. Y.-C., Wei, T.-C., Zhang, C., Yu, H., & Yoo, S. (2022). Quantum convolutional neural networks for high energy physics data analysis. *Physical Review Research*, *4*(1). doi: 10.1103/physrevresearch.4.013231

Chen, Z., Du, C., Huang, L., Li, D., & He, H. (2018). Improving Image Classification Performance with Automatically Hierarchical Label Clustering. *2018 24th International Conference on Pattern Recognition (ICPR)*. doi: 10.1109/icpr.2018.8546042

Dai, Z., Liu, Y., Di, S., & Fan, Q. (2021). Aspect-level sentiment analysis merged with knowledge graph and graph convolutional neural network. *Journal of Physics: Conference Series*, *2083*(4), 042044. doi: 10.1088/1742-6596/2083/4/042044

Deng, J., Li, K., Do, M., Su, H., & Fei-Fei, L. (2009). Construction and Analysis of a Large Scale Image Ontology..

Dharani, T., & Laurence Aroquiaraj, I. (2017). An Essential Image Augmentation Processes for Pattern Based Image Retrieval System. *SSRN Electronic Journal*. doi: 10.2139/ssrn.3120213

Divya Meena, S., & Agilandeeswari, L. (2019). An Efficient Framework for Animal Breeds Classification Using Semi-Supervised Learning and Multi-Part Convolutional Neural Network (MP-CNN). *IEEE Access*, *7*, 151783–151802. doi: 10.1109/access.2019.2947717

Education, I. C. (2020, 10). *Gradient Descent.* Retrieved from `https://www.ibm.com/cloud/learn/gradient-descent`

Finlayson, G. D. (2018). Colour and illumination in computer vision. *Interface Focus*, *8*(4), 20180008. doi: 10.1098/rsfs.2018.0008

Frei, M., & Kruis, F. (2021). FibeR-CNN: Expanding Mask R-CNN to improve image-based fiber analysis. *Powder Technology*, *377*, 974–991. doi: 10.1016/j.powtec.2020.08.034

Khan, L. (2007). Standards for image annotation using Semantic Web. *Computer Standards Interfaces*, *29*(2), 196–204. doi: 10.1016/j.csi.2006.03.006

Kontopoulos, E., Berberidis, C., Dergiades, T., & Bassiliades, N. (2013). Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, *40*(10), 4065–4074. doi: 10.1016/j.eswa.2013.01.001

Li, W. (2021, mar). Analysis of object detection performance based on faster r-CNN. *Journal of Physics: Conference Series*, *1827*(1), 012085. Retrieved from `https://doi.org/10.1088/1742-6596/1827/1/012085` doi: 10.1088/1742-6596/1827/1/012085

Marr, B. (2021, 12). *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read.* Retrieved from `https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/?sh=6d6da4fb60ba`

Meng, Y., Wang, G., & Liu, Q. (2019). Multi-layer Convolutional Neural Network Model Based on Prior Knowledge of Knowledge Graph for Text Classification.

2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). doi: 10.1109/icccbda.2019.8725669

Menglong, C., Detao, J., Ting, Z., Dehai, Z., Cheng, X., Zhibo, C., & Xiaoqiang, X. (2019). Image Classification Based on Image Knowledge Graph and Semantics. *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. doi: 10.1109/cscwd.2019.8791922

Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of Deep Learning for Object Detection. *Procedia Computer Science*, *132*, 1706–1717. doi: 10.1016/j.procs.2018.05.144

Paulauskaite-Taraseviciene, A., Sutiene, K., Valotka, J., Raudonis, V., & Iesmantas, T. (2019). Deep Learning-based Detection of Overlapping Cells. *Proceedings of the 2019 3rd International Conference on Advances in Artificial Intelligence*. doi: 10.1145/3369114.3369120

Pellegrini, T. (2016, 09). *Inferring phonemic classes from CNN activation maps using clustering techniques.* Retrieved from `https://hal.archives-ouvertes.fr/hal-01474886`

Rashu, S., Ghule, V., Chate, R., Chinchnae, P., & Ghodke, S. (2021). Vehicle Classification using CNN. *Strad Research*, *8*(5). doi: 10.37896/sr8.5/078

Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2019, 09–15 Jun). Do ImageNet classifiers generalize to ImageNet? In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 5389–5400). PMLR. Retrieved from `https://proceedings.mlr.press/v97/recht19a.html`

Roy, D., Panda, P., & Roy, K. (2020). Tree-CNN: A hierarchical Deep Convolutional Neural Network for incremental learning. *Neural Networks*, *121*, 148–160. doi: 10.1016/j.neunet.2019.09.010

Rubin, D. L., Mongkolwat, P., Kleper, V., Supekar, K., & Channin, D. S. (2009, 1). Annotation and Image Markup: Accessing and Interoperating with the Semantic Content in Medical Imaging. In (pp. 57–65). Institute of Electrical and Electronics Engineers (IEEE). doi: 10.1109/mis.2009.3

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning Integral Representation by Error Propagation. *Institute of Cognitive Science*(8506).

Shi, L., Li, S., Yang, X., Qi, J., Pan, G., & Zhou, B. (2017). Semantic Health Knowledge Graph: Semantic Integration of Heterogeneous Medical Knowledge and Services. *BioMed Research International*, *2017*, 1–12. doi: 10.1155/2017/2858423

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1). doi: 10.1186/s40537-019-0197-0

Smith, S., jan Kindermans, P., Ying, C., & Le, Q. V. (2018). Don't decay the learning rate, increase the batch size.. Retrieved from `https://openreview.net/pdf?id=B1Yy1BxCZ`

Sugata, T. L. I. (2017, 11). *Architecture of VGG16.* Retrieved from `https://www.researchgate.net/publication/321829624/figure/fig2/AS:571845657481217@1513350037610/VGG16-architecture-16.png`

Wang, D., & Mao, K. (2019). Learning Semantic Text Features for Web Text-Aided Image Classification. *IEEE Transactions on Multimedia*, *21*(12), 2985–2996. doi: 10.1109/tmm.2019.2920620

Yumeng, Z., Jing, Y., Shuo, G., & Limin, L. (2021). News Image-Text Matching With News Knowledge Graph. *IEEE Access*, *9*, 108017–108027. doi: 10.1109/access.2021.3093650