HASAN KALYONCU UNIVERSITY

GRADUATE EDUCATION INSTITUTE

CLUSTERING METHODS FOR LARGE SCALE VISUAL PLACE RECOGNITION

ELECTRONICS AND COMPUTER ENGINEERING

M. Sc. THESIS

BY

AHMAD EL JOUMA

JAN 2023

January 2023

M.Sc. in ELECTRONICS AND COMPUTER ENGINEERING AHMAD EL JOUMA

CLUSTERING METHODS FOR LARGE SCALE VISUAL PLACE RECOGNITION

Electronics and Computer Engineering

Hasan Kalyoncu University

M.Sc. Thesis

Supervisor

Assoc. Prof. Dr. Abdul Hafiz ABDULHAFIZ

By

Ahmad EL JOUMA

JAN 2023

	GRADUATE EDUCATION INSTITUTE **M.Sc. ACCEPTANCE AND APPROVAL FROM**

Electronics-Computer Engineering M.Sc. (Master Of Science) program student **Ahmad EL JOUMA** prepared and submitted the thesis titled " **CLUSTERING METHODS FOR LARGE SCALE VISUAL PLACE RECOGNITION**" defended successfully on the date of 19/01/2023 and accepted by the jury as an M.Sc. thesis.

Position	**Title, Name and Surename** **Department/University**	**Signature:**
M.Sc. Supervisor	Assoc. Prof. Dr. ABDUL HAFIZ ABDULHAFIZ Computer Engineering Department Hasan Kalyoncu University	
Jury Member	Assoc. Prof. Dr. BÜLENT HAZNEDAR Electrical and Electronics Engineering Department Gaziantep University	
Jury Member	Asst. Prof. Dr. Saed ALQARALEH Computer Engineering Department Hasan Kalyoncu University	

This thesis is accepted by the jury members selected by institute management board and approved by institute management board.

Prof.Dr. M. Serhat YENİCE

Enstitü Müdürü

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ahmad EL JOUMA

Signature

ABSTRACT

CLUSTERING METHODS FOR LARGE SCALE VISUAL PLACE RECOGNITION

EL JOUMA, Ahmad
M.Sc. in Electroincs and Computer Eng.
Supervisor: Assoc. Prof. Dr. Abdul Hafiz ABDULHAFIZ
Jan 2023
84 pages

In this thesis, a new solution for the problem that most Visual Place Recognition systems suffer from, which is directly proportional between the size of searching community and time. The proposed solution is based on the clustering principles, where clustering in general works on grouping the samples (images datasets in our case) based on some criteria like file similarity.

Searching in a large-scale image database is a complex problem because the searching operations in the whole database take a large amount of time, and comparing calculations is so critical to be done as fast as possible and extract the best (most relevant) results from the whole database.

In order to build the system, we first compared multiple feature extraction methods like AlexNet CNN response features, SIFT and HOG features. Also, the Manhattan distance was utilized for measuring the similarity(distances) between the features. Furthermore, extensive comparisons for AlexNet and RESNET-18 have been done. Overall, RESNET-18 outperformed AlexNet, although AlexNet response time is shorter.

Two components are the main parts of the proposed visual Place Recognition system. The first part(component) consisted of one RESNET-18 model, while the second part has multiple copies of RESNET-18. Allocating the best cluster to which the enquired image belongs is the goal of the first component. While finding the ID of the requested image is the second part's obligation.

The proposed system in this thesis shows superior performance over many state-of-the-art methods on several challenging benchmark datasets. And was able to obtain a State-of-the-art result execution time, where the total time to produce the result of the inquired image, was between 200-220 ms.

Key Words: ALEXENET CNN, K-MEANS, SIFT, HOG, RESNET, VPR.

ÖZET

BÜYÜK ÖLÇEKLİ GÖRSEL YER TANIMA İÇİN KÜMELEME YÖNTEMLERİ

EL JOUMA, Ahmad
Yüksek Lisans Tezi, Elektronik-Bilgisayar Mühendisliği

Tez Yöneticisi: Doç. Prof. Abdul Hafiz ABDULHAFIZ

Ocak 2023,

84 sayfa

Bu tezde, çoğu Görsel Yer Tanıma sisteminin muzdarip olduğu, arama topluluğunun boyutu ve zaman arasında doğru orantılı olan soruna yeni bir çözüm getirilmiştir. Önerilen çözüm, kümelemenin genel olarak örnekleri (bizim durumumuzda görüntü veri kümeleri) dosya benzerliği gibi bazı kriterlere göre gruplandırmaya çalıştığı kümeleme ilkelerine dayanmaktadır.

Büyük ölçekli bir görüntü veritabanında arama yapmak karmaşık bir sorundur çünkü tüm veritabanındaki arama işlemleri çok zaman alır ve karşılaştırma hesaplamaları olabildiğince hızlı yapılmak ve veri tabanından en iyi (en alakalı) sonuçları çıkarmak için çok önemlidir. tüm veritabanı.

Sistemi kurmak için öncelikle AlexNet CNN yanıt özellikleri, SIFT ve HOG özellikleri gibi çoklu özellik çıkarma yöntemlerini karşılaştırdık. Ayrıca, özellikler arasındaki benzerliği (uzaklıkları) ölçmek için Manhattan mesafesi kullanılmıştır. Ayrıca, AlexNet ve RESNET-18 için kapsamlı karşılaştırmalar yapılmıştır. Genel olarak, RESNET-18, AlexNet yanıt süresi daha kısa olmasına rağmen, AlexNet'ten daha iyi performans gösterdi.

İki bileşen, önerilen görsel Yer Tanıma sisteminin ana parçalarıdır. Birinci kısım(bileşen) bir RESNET-18 modelinden oluşurken, ikinci kısım RESNET-18'in birden çok kopyasına sahiptir. Sorgulanan görüntünün ait olduğu en iyi kümeyi tahsis etmek, birinci bileşenin amacıdır. İstenen görselin ID'sini bulmak ise ikinci kısmın yükümlülüğündedir.

Bu tezde önerilen sistem, birkaç zorlu kıyaslama veri setinde en son teknolojiye sahip birçok yönteme göre üstün performans göstermektedir. Ve sorgulanan görüntünün sonucunu üretmek için toplam sürenin 200-220 ms arasında olduğu son teknoloji bir sonuç yürütme süresi elde edebildi.

Anahtar Kelimeler: ALEXENET CNN, K-MEANS, SIFT, HOG, RESNET, VPR.

ACKNOWLEDGMENTS

I would like to express my special thanks of gratitude to my supervisor Dr. Abdul Hafiz ABDULHAFIZ who gave me the golden opportunity to do this wonderful project with nice notes and instructions. I would also thank Tübitak and Hasan Kalyoncu Üniversitesi for their cooperative and support. I would also thank Prof. Turkey Dereli and the head of electroincs and computer engineering M.Sc. program Prof. Dr. M. Fatih HASOĞLU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGRURES

LIST OF SYMBOLS

AUC	Area Under The Curve
ADC	Asymmetric Distance Computation
BOF	Bag Of Features
BoW	Big of Words
BRIEF	Binary Robust Independent Elementary Features
CNN	Convolutional Neural Network
FAB-MAP	Probabilistic Localization and Mapping in the Space of Appearance
FAST	Features from Accelerated Segment Test
FPR	False Positive Rate
FV	Fisher Vectors
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
LDA	Linear Discriminant Analysis
LSVRC	Large Scale Visual Recognition Challenge
LTSM	Long Short-Term Memory
OSS	One-Shot Similarity
PCA	Principal Component Analysis
QUT	Queensland University of Technology
ROC	Receiver Operating Characteristic
SAD	Sum of Absolute Differences
SIFT	Scale Invariant Feature Transform
SSE	Sum of squared errors
SURF	Speeded-Up Robust Features
TPR	True Positive Rate
VLAD	Vector of Locally Aggregated Descriptor
VPR	Visual Place Recognition

CHAPTER 1

1. INTRODUCTION

The procedure of locating the robot is called "visual place recognition," or VPR. VPR is sometimes viewed as an image retrieval system that compares an image to places in a stored database in an effort to identify a location (Khaliq et al., 2019). The issue with VPR is now the issue with searching a database of pictures for a query image.

The method suggested in this thesis is dedicated to large-scale images database. When the size of the image database becomes larger and larger, sequential searching becomes without benefits and takes much time which is not appropriate for real-time systems (Facil et al., 2019). Our suggested method depends on CNN as a backbone for the search process. The suggested method has shown state-of-art results and outperforms other state-of-art methods while significantly decreasing execution time.

As a starting point to solve the sequential searching problem, clusters of images should be built. Clustering of images is the process of grouping samples depending on similarity and spatial distances, In this work medium size for clustering is chosen, as clusters sizes may vary between large to small sizes, given that small size will make the system overloading, as there will be a large number of clusters to deal with. On the other hand, the large cluster makes CNN overloading, as each cluster will have a huge number of images that should be distinguished.

1.1 Literature Review

The VPR research domain has acquired a great amount of attention in recent decades. Early research investigations used locally created feature descriptors (Garg et al., 2022) such as SIFT (Lowe, 2004), SURF (Bay et al., 2006), or BRIEF (Calonder et al., 2010), where the Scale Invariant Feature Transform (SIFT) method converts image data into scale-invariant coordinates concerning local features (Lowe, 2004). This method's ability to produce many features that densely cover the image at all sizes and locations is a key component (Lowe, 2004). SIFT characteristics are initially taken from a set of

reference photos and stored in a database for image matching and recognition (Lowe, 2004). A fresh picture is matched by individually comparing each feature from the new image to this prior database and determining potential matching features based on the Euclidean distance of their feature vectors (Lowe, 2004).

The Hessian matrix is the basis of SURF (Speeded-Up Robust Features) (Bay et al., 2006). To shorten the computing time, it uses integral pictures (Bay et al., 2006). In the neighbourhood surrounding the interesting point, the descriptor specifies a distribution of Haar-wavelet responses (Bay et al., 2006). To increase speed, they used integrated pictures (Bay et al., 2006). A further benefit of using only 64 dimensions is that it speeds up robustness while also cutting down on the time required for feature calculation and matching (Bay et al., 2006). The process for matching looks like this, calculating the Euclidean distance between their descriptor vectors allows us to compare an interesting point in the test image to an interesting point in the reference image (Bay et al., 2006). If the distance between the two is less than 0.7 times that of the second closest neighbour, a matching pair is found (Bay et al., 2006).

BRIEF is an effective feature point descriptor that uses binary strings (Calonder et al., 2010). They demonstrated that it can be calculated using straightforward intensity difference tests and is highly discriminative even when utilizing only a few bits (Calonder et al., 2010). Additionally, rather than utilizing the L2 norm as is typically done, the Hamming distance, which is exceedingly efficient to compute, may be used to evaluate the descriptor similarity (Calonder et al., 2010). The BRIEF method never generates a long descriptor but instead directly constructs small ones by comparing the intensities of pairs of points (Calonder et al., 2010).

Other methods employed global feature descriptors such as Fisher Vectors (FV) (Jégou et al., 2011), GIST (Oliva, 2005), or HOG (Fazl-Ersi and Tsotsos, 2012). By encoding high-order statistics, the Fisher vector enhances the BOW (first and, optionally, second order) (Jégou et al., 2011). This description vector, scaled by the inverse square root of the Fisher information matrix, represents the gradient of the sample's probability with respect to the parameters of this distribution (Jégou et al., 2011). When compared to a parametric generative model's average descriptor distribution, FV shows how the

collection of descriptors deviates from that model (Jégou et al., 2011). For the purpose of contrasting the vector approximations the dimensionality reduction and the quantization, respectively, were introduced (Jégou et al., 2011).

GIST is a visual system that creates a rich enough spatial picture of the outside world during this quick process of seeing, and detecting a few objects and other relevant information in the image, to aid in object identification and the deployment of attention (Oliva, 2005). The essence of a scene is referred to by this representation, which includes all processing levels, from low-level details (such as colour and spatial frequencies) to intermediate image characteristics (such as surface and volume) and high-level details (such as objects and activation of semantic knowledge) (Oliva, 2005). As a result, both perceptual and intellectual levels may be used to study gist. Information that is highly salient yet unclear may be more useful for a first approximate estimation of the scene's essence if a scene is unfamiliar and must be swiftly classified (Oliva, 2005).

Histogram of Oriented Gradient (HOG) offers a substantial degree of generalization for location classification while delivering a strong level of discriminative power for place identification (Fazl-Ersi and Tsotsos, 2012). In a subdivision framework, HOG descriptors are employed to represent images, and an informative feature selection approach based on kernel alignment is utilized to determine the size and position of sub-regions (Fazl-Ersi and Tsotsos, 2012). A variation of the One-Shot Similarity (OSS) metric is used in HOG to try to develop resilience to perceptual aliasing (Fazl-Ersi and Tsotsos, 2012). A model for each of the two HOG descriptors is learned using the Linear Discriminant Analysis (LDA) algorithm given a pair of descriptors (Fazl-Ersi and Tsotsos, 2012). For the purpose of calculating a probability score, each of the two trained models is applied to the other descriptor (Fazl-Ersi and Tsotsos, 2012). To determine how similar the two descriptions are overall, the two projected scores are then added together (Fazl-Ersi and Tsotsos, 2012).

Bag of Word (BoW) (Nister and Stewenius, 2006) strategy is resistant to background occlusion and background clutter and draws on well-liked indexing descriptor extraction approaches from local areas (Nister and Stewenius,2006). In a vocabulary tree, the descriptors of the local region are quantized hierarchically (Nister and Stewenius,

2006). An expansive and discriminating vocabulary may be used effectively with the help of the vocabulary tree (Nister and Stewenius, 2006). The tree directly specifies the quantization, which is the scheme's most important characteristic (Nister and Stewenius, 2006). Due to their complete integration, quantization and indexing are effectively the same (Nister and Stewenius, 2006). By using hierarchical k-means clustering, the word tree defines a hierarchical quantization (Nister and Stewenius, 2006). During the tree's unsupervised training, a sizable collection of representative descriptor vectors is employed (Nister and Stewenius, 2006). After the quantization is established, in order to assess the similarity of the vocabulary tree pathways leading to the descriptors from the database picture and the query image to assess the relevance of a database image to the query image (Nister and Stewenius, 2006).

In DBoW (Gálvez-López and Tardos, 2012), using a bag of words derived from FAST+BRIEF characteristics, they suggested a unique technique for visual location detection (Gálvez-López and Tardos, 2012). They created a language tree that discretizes a space of binary descriptors, then utilized the tree to accelerate correspondences for geometrical verification (Gálvez-López and Tardos, 2012). The entire method—including feature extraction—takes 22 milliseconds per frame in a sequence of 26300 photos, which is an order of magnitude quicker than prior methods (Gálvez-López and Tardos, 2012). They proposed a method to avoid photographs gathered recently and representing the same location from competing with each other during the matching process and made use of a bag of binary words for the first time (Gálvez-López and Tardos, 2012). This allowed them to operate at a greater frequency (Gálvez-López and Tardos, 2012). They employed a hierarchical bag of words, direct and inverse indexes, and a picture database to identify locations that had been visited previously (Gálvez-López and Tardos, 2012).

In FAB-MAP (Cummins and Newman, 2008) they employed a "bag-of-words" model of raw sensor data in which situations are represented as a series of characteristic words selected from a predetermined lexicon (Cummins and Newman, 2008). They used binary characteristics from images that were based on quantized SURF descriptors, but binary features from any sensor or group of sensors could be used (Cummins and Newman, 2008).

In the Vector of Locally Aggregated Descriptor (VLAD) (Jégou et al., 2010) technique, they tackled the issue of picture search on a very large scale, where three limitations must be taken into account simultaneously: the search's accuracy, efficiency, and memory utilization of the representation (Jégou et al., 2010). They originally suggested a straightforward method for grouping local image descriptors into a vector with a finite number of dimensions, which may be seen as a simplification of the Fisher kernel representation (Jégou et al., 2010).VLAD involves offering a representation that offers great search accuracy with manageable vector dimensionality (Jégou et al., 2010). They suggested a descriptor that combines SIFT descriptors and creates a compact representation. This descriptor was generated using Bag Of Features (BOF) and Fisher kernel. To search in a large-scale database, they used an approach that incorporates the vector into a binary space that better meets the memory restriction (Jégou et al., 2010). The product quantization-based approximation search approach, however, greatly outperforms it in terms of the trade-off between memory and accuracy. In doing so, they can contrast the vector approximations that the dimensionality reduction and quantization introduced (Jégou et al., 2010). They applied the form of this strategy known as asymmetric distance computation (ADC), which only encrypts the database's vectors (Jégou et al., 2010).

However, when compared to Deep Learning methods, these findings are not precise enough (Khaliq et al., 2019). Many researchers used CNN to solve VPR problems (Krizhevsky et al., 2017). Krizhevsky et al. proposed a CNN to categorize 1.2 million high-resolution photographs entered in the ImageNet LSVRC-2010 competition, into 1000 distinct classes (Krizhevsky et al., 2017). Their outcome outperforms the prior state-of-the-art by a significant margin (Krizhevsky et al., 2017). The CNN consists of three fully connected layers, five convolutional layers, some of which are followed by max-pooling layers, and a final 1000-way softmax layer (Krizhevsky et al., 2017). It includes 60 million parameters and 650,000 neurons (Krizhevsky et al., 2017). Using a recently established regularization technique dubbed "dropout," they were able to significantly minimize overfitting in the fully linked layers (Krizhevsky et al., 2017).

According to recent findings, CNN's' generic descriptors are quite effective (Sharif et al.,2014). Sharif et al. employed features derived from the OverFeat network as a general image representation to take on a variety of identification tasks, including object image classification, scene recognition, fine-grained recognition, attribute detection, and image retrieval applied to a variety of datasets (Sharif et al., 2014). These tasks and datasets were chosen because they gradually deviated from the ones the OverFeat network was trained on, which is how they were chosen. Surprisingly, they consistently outperformed highly calibrated state-of-the-art systems on a variety of visual categorization tasks (Sharif et al., 2014). Layer 22 of the network, the top completely linked layer, serves as their feature vector (Sharif et al., 2014). They concluded that CNN-based deep learning should be the top choice for virtually all image identification tasks (Sharif et al.,2014).

Razavian et al. conducted a thorough investigation on the availability of convolutional network-based picture representations to retrieve visual instances (Razavian et al., 2016). Along with the selection of convolutional layers, they also provided an effective pipeline that extracts local features by utilizing multi-scale schemes and explicitly accounting for geometric invariance, that is, locations, scales, and spatial consistency (Razavian et al., 2016). They proved in their studies using five common image retrieval datasets that generic ConvNet picture representations may outperform other cutting-edge techniques if they are extracted properly (Razavian et al., 2016).

Sünderhauf et al. offered a method for identifying prospective landmarks in an image for location recognition by using cutting-edge object suggestion algorithms (Sünderhauf et al., 2015). They performed location identification over a wide range of appearance and perspective alterations using the astounding capacity of convolutional neural network features to find matching landmark suggestions between photos (Sünderhauf et al., 2015). Their system does not require any kind of training because all of its parts are generic enough to be purchased without a prescription (Sünderhauf et al., 2015). It computes features using an off-the-shelf pre-trained CNN and extracts landmark suggestions from pictures using a generic object proposal system (Sünderhauf et al., 2015).In the photos, they extracted recognized and steady areas that naturally tend to serve

as trustworthy markers (Sünderhauf et al., 2015). The resilience against viewpoint changes or partial occlusions in the scenes is considerably improved by using landmark sections rather than the entire picture to represent a scene (Sünderhauf et al., 2015). ConvNet characteristics have been demonstrated to be more resistant to changes in look and condition than previous techniques (Sünderhauf et al., 2015).

Chen et al. proposed two CNN architectures (AMOSNet and HybridNet) that were trained for the particular location identification task, and they used a multi-scale feature encoding technique to create features that are condition- and viewpoint-invariant (Chen et al., 2017). On several difficult benchmark place identification datasets, they thoroughly assess their trained networks and show that they perform on par with pre-trained Convolution networks and other place recognition algorithms, with an average performance improvement of 10% (Chen et al., 2017). They showed that on all network levels, AMOSNet and HybridNet provide features that are more resistant to changes in appearance and point of view than CaffeNet.Layers Conv4, Conv5, and Conv6 are where the HybridNet creates the most strong features, while Layers fc7 and fc8 are where the PlaceNet excels over all other networks (Chen et al.,2017). The conv4 layer, generated by HybridNet, has the greatest performance (Chen et al., 2017).

Arandjelovic et al. for the purpose of performing the location recognition job, a CNN model that is trainable end-to-end was created. NetVLAD, a novel generalized VLAD layer, serves as the foundation of this design (Arandjelovic et al., 2016). They created a training method based on a novel weakly supervised ranking loss to learn the model's parameters from beginning to end using pictures of the same locations taken throughout time and retrieved from Google Street View Time Machine (Arandjelovic et al., 2016). They demonstrated that, on two difficult place identification benchmarks, the proposed model greatly outperforms non-learnt image representations and commercial Convolution Network descriptors, and outperforms state-of-the-art compact image representations on common picture retrieval benchmarks (Arandjelovic et al., 2016). A huge geotagged picture database is visually searched using a query image with an unknown location, and the locations of the highest-ranking photos are utilized as recommendations for the query's location (Arandjelovic et al., 2016). Typically, this is

accomplished by creating a function f that serves as the picture representation extractor (Arandjelovic et al., 2016).

Chen et al. utilized a spatial and sequential filter in conjunction with the significant features learned by CNNs (Chen et al., 2014). Using the technique to navigate a 70 km dataset for location recognition, they acquired a 75% substantial increase in the recall at 100% accuracy surpassing all prior state-of-the-art methods (Chen et al., 2014). The element with the smallest feature vector difference is sought for in each column in the confusion matrix to determine the strongest place match hypothesis (Chen et al., 2014). A more complex version of SeqSLAM's basic motion filter, the sequential filter is applied. It is worth noting that various layers tend to be better suited to different parts of the location recognition test, with the middle layers doing best on relatively static, comparable perspective datasets, and later layers performing better when viewpoint diversity becomes large (Chen et al., 2 014). A straightforward Euclidean distance metric is used to measure the similarity of feature responses, a method that implicitly implies that each feature contributes equally to place identification performance (Chen et al., 2014).

Olid et al. showed that despite appearance variations, they trained a CNN to extract feature vectors that are comparable to those acquired from photographs of the same location (Olid et al., 2018). Euclidean distance is used as a measurement (Olid et al., 2018). They accepted that the feature space distribution, loss, and convergence of optimization are all significantly influenced by the distance function (Olid et al., 2018). By simultaneously training positive and negative pairings, triplet neural networks enhance the siamese model’s output (Olid et al., 2018). A more consistent and effective learning process results from moving descriptors from the same location apart and farther away in the same instant from descriptors from other locations (Olid et al., 2018). They came to the conclusion that the fine-tuned triplet network, which started with the weights of the pre-trained VGG-16-Places and added a fully-connected layer with an output dimension of 128, produced the greatest results (Olid et al., 2018).

Lowry and Milford showed that under a variety of place identification scenarios, they suggested an unsupervised change removal approach and compared it to a traditional supervised change prediction learning technique (Lowry and Milford, 2016). They

demonstrated that change removal beats change prediction and that even when trained with the incorrect appearance circumstances, change removal may enhance place identification performance by up to 900% whereas change prediction fails dramatically when given the incorrect training data (Lowry and Milford, 2016). Change removal has also been demonstrated to outperform state-of-the-art location identification systems that employ illumination invariant pictures and convolution network features when trained with as low as 100 samples (Lowry and Milford, 2016). Many visual place identification systems can incorporate change removal (Lowry and Milford, 2016). To figure out how to transition between different appearance conditions, they employed linear regression (Lowry and Milford, 2016). Principal component analysis (PCA) helps identify and eliminate a setting's common elements (Lowry and Milford, 2016).

Facil et al. put out three potential solutions (Descriptor Grouping, Fusion, and Recurrent Descriptors) that deep networks may use to merge the visual properties of several frames in a sequence (Facil et al., 2019). In two open datasets, they demonstrated that their methods yielded more compact and effective descriptions than the single- and multi-view baselines in the literature (Facil et al., 2019). They used ResNet-50 as the foundation (Facil et al., 2019). It was trained on ImageNet (Facil et al., 2019). They aimed to create a model that could learn to combine the data from an n-frames window into a 128-dimensional descriptor that was both more discriminating and smaller (Facil et al., 2019). To learn how to integrate the outputs of many ResNet-50 models into a single, compact descriptor, they created an additional fully connected layer using the Descriptor Fusion technique (Facil et al., 2019). This network can represent more complicated instances since it can learn how to weigh the information from various frames (Facil et al., 2019). Descriptor Grouping, for instance, has limitations when sequences are recorded in reverse order, but Descriptor Fusion can learn an appropriate fusion (Facil et al., 2019). They desired to maintain the most pertinent prior information while updating the sequence descriptor live as new frames arrived (Facil et al., 2019). They suggested a recurrent neural network to do it. In their model, the top layers feed an LSTM network, which creates a 128-dimensional descriptor, and every frame feeds a ResNet-50 (Facil et al., 2019). Each input frame updates the inner state that LSTMs maintain, and the output is dependent on both the state and the input (Facil et al., 2019). In contrast to other models, their network

can create a descriptor from the initial frame and update it sequentially when further frames are received by maintaining a recurrent inner state (Facil et al., 2019).

Garg et al. suggested a descriptor called Delta Descriptors, which is formed by observing changes in any taught global descriptor over time (Garg et al., 2020). By taking into account temporal variations between locations seen along a route, delta descriptors reduce the offsets caused in the initial descriptor matching space in an unsupervised way (Garg et al., 2020). Delta Descriptors, like all other systems, have a drawback called volatility on a frame-by-frame basis that may be solved by combining them with sequential filtering techniques (Garg et al., 2020). They showed the great performance of Delta Descriptors alone using two benchmark datasets, before demonstrating new state-of-the-art performance when paired with sequence-based matching. Additionally, they provided evidence that the method worked with four distinct underlying descriptor types as well as two additional advantages of Delta Descriptors over competing methods: an improved inherent robustness to variations in camera motion and a slower rate of performance degradation as the dimensional reduction was used (Garg et al., 2020). They put forth an unsupervised technique called Delta Descriptors for converting current deep-learnt image descriptors into change-based representations (Garg et al., 2020). The initial descriptors often tend to be offset by a margin due to changes in the environment during a revisit, increasing the distance between descriptors referring to the same location (Garg et al., 2020). Delta Descriptors, which are defined in a different space, inherently account for such offsets, resulting in a smaller separation between descriptors for the same location (Garg et al., 2020). This is crucial for mobile robots or autonomous vehicles working in unfamiliar environments since they may experience major changes in their visual appearance while traversing the same path repeatedly (Garg et al., 2020). They found that the suggested Delta Descriptors outperform sequence-based matching when given a fixed sequential span (Garg et al., 2020).

Baumgartl and Buettner. offered a model for indoor place recognition that was extremely accurate and reliable (Baumgartl and Buettner., 2020). Utilizing a pre-trained deep convolutional neural network and the explicit inductive bias transfer learning approach, a complete end-to-end convolutional neural network architecture was created

(Baumgartl and Buettner., 2020). Excellent performance values are demonstrated by experimental findings using data from York University and Rzeszow University (Baumgartl and Buettner, 2020). Superior performance to current benchmarks in terms of resilience against changes in camera angle and illumination (Baumgartl and Buettner, 2020). Additionally, compared to the benchmark, their design is 82.46 percent smaller (Baumgartl and Buettner, 2020). The MobileNetV2 model was used (Baumgartl and Buettner, 2020). A typical method to improve classification performance and reduce model overfitting is data augmentation (Baumgartl and Buettner, 2020). Their recognition module makes use of a MobileNetV2-based transfer learning method with an explicit inductive bias transfer learning strategy, a 1.0 width multiplier, and input dimensions of 224x224 pixels (Baumgartl and Buettner, 2020). Depending on the dataset used, the MobileNetV2 model's final classification layer has been replaced to match the number of classes in each dataset (Baumgartl and Buettner, 2020).

Hausler et al. stated clearly that the use of multisensor fusion and the integration of information collected over time from picture sequences are two common methods used to increase the effectiveness of visual location identification algorithms (Hausler et al., 2019). Although these methods can increase performance, they have drawbacks such as the requirement for additional physical sensors and calibration procedures for both multiple sensors and the image-matching sequence duration (Hausler et al., 2019). Their method imposes a set of requirements on the image processing procedures: instead of relying on a single, optimally performing method, their multi-approach method only needs the various processing methods to work well together across a variety of deployed environments, which lowers the performance requirements of any single processing method (Hausler et al., 2019). They incorporated four distinct picture-processing strategies that have been shown in the literature to perform in a variety of ways (Hausler et al., 2019).

1. Patch normalization with Sum of Absolute Differences (SAD) (Hausler et al., 2019).
2. Using a feature map, extract convolution network features from several spatial areas (Hausler et al., 2019).

3. Histogram of Oriented Gradients (HOG) (Hausler et al., 2019).
4. Convolution network features, but solely makes use of the spatial most activations (Hausler et al., 2019).

They suggested an approach for integrating many techniques for image processing utilizing a series of images with the Hidden Markov Model (HMM) (Hausler et al., 2019). Finding the best sequence sub-set to utilize in the HMM by generating a dynamic sequence matching length that measures the rate of change in recognition quality over a sample of recent photos (Hausler et al., 2019). Multi-Process Fusion is a technique for automatically choosing the most effective image processing techniques for the present situation (Hausler et al., 2019). They employed the specialized Viterbi method to establish the predicted location and duration of a dynamic sequence to assess place matches using a new trajectory variation grading system utilized by SeqSLAM (Hausler et al., 2019). They have created a final component that employs a voting mechanism to assess the matching performance of each processing technique separately and eliminate the one that performs the worst (Hausler et al., 2019).

Chen, Maffra and et al. by recognizing salient regions and producing their regional representations directly from the convolutional layer activations, they presented CNN-based visual characteristics for location detection (Chen, Maffra and et al., 2017). They shed light on what makes a picture presentation resilient to outside changes (Chen, Maffra and et al., 2017). An approach for feature encoding based on a convolutional neural network makes visual representations that let the explanation of several picture areas without having to provide several inputs to the Convolutional network model a technique for visual location identification that is based on regions can handle differences in perspective and conditions, concurrently (Chen, Maffra and et al., 2017). Firstly they have done is the extraction of local descriptors using convolutional activations. The purpose of this stage is to derive local representations for a specific area of a picture straight from the convolutional layer activations. For this phase, they simply use a pre-trained convolutional neural network, taking into account its convolutional layers and completely discarded linked layers (Chen, Maffra and et al., 2017). Convolutional layer activations typically incorporate significant spatial information, from which local descriptors may be generated

(Chen, Maffra, et al., 2017). This is their primary benefit over completely linked ones (Chen, Maffra, et al., 2017). The preceding step's feature extraction can only describe an area as large as its filter's receptive field (Chen, Maffra, et al., 2017). However, with place identification, each site is represented by a collection of recognizable landmarks, which can be any size and form (Chen, Maffra, et al., 2017). A simple approach would be to combine all local descriptors falling into that region to produce a pooled feature vector, which could then be used to describe a visual pattern at different sizes and in arbitrary shapes (Chen, Maffra, et al., 2017). They immediately mined distinguishing patterns from a late convolutional layer to find landmarks relevant for location recognition (Chen, Maffra, et al., 2017). In general, the detection scores obtained by using the convolution filter on the input picture may be used to interpret a feature map produced by a convolutional filter (Chen, Maffra, et al., 2017). High activation values on this map imply the presence of nearby visual patterns that the filter is looking for. Feature maps at a late convolutional layer are seen to be typically relatively sparse and selective to visual patterns relating to certain semantically important regions, such as a form or an object portion (Chen, Maffra, et al., 2017). They suggested a landmark finding method based on these hypotheses, looking for the strongest spatially isolated patches at late convolutional feature maps. Each pooling vector describes a single area of the picture that the pre-trained model deems significant (Chen, Maffra, et al., 2017). They devised a method to determine the inverse-document frequency of each of these key areas (Chen, Maffra, et al., 2017). Cross-matching was done between all area vectors that were extracted from both photos to compare the similarity between the two images (Chen, Maffra, et al., 2017). The database's reference photos are all reviewed in order to get the one with the greatest similarity score that best matches the query image (Chen, Maffra, et al., 2017).

Khaliq et al. provided a lightweight solution for visual location recognition that can perform well with little computing expense and is practical for mobile robots with frequent changes in perspective and appearance (Khaliq et al., 2019). Their results from numerous benchmark datasets show a 13% increase in accuracy and a 12x increase in speed when compared to cutting-edge approaches (Khaliq et al., 2019). To lower the memory and computational costs, they suggested a comprehensive strategy aimed at a convolutional neural network model with a few layers that were pre-trained on scene-

centric picture databases (Khaliq et al., 2019). Their suggested approach finds important regional characteristics based on a convolutional neural network and combines them with a vector of the locally aggregated descriptor (VLAD) that has been specially tailored to the visual place problem (Khaliq et al., 2019). The reason for using VLAD is that it performs better than BoW in a variety of CNN-based image retrieval tasks while using a smaller visual word vocabulary (Khaliq et al., 2019). The first effort combines brand-new, lightweight regional features based on a convolutional neural network with VLAD encoding customized for the visual place (Khaliq et al., 2019). The method used the middle convolutional layer of the eight-layered convolutional neural network model, which provided improved accuracy (Khaliq et al., 2019). In terms of AUC computed under precision-recall curves, estimation on numerous viewpoint- and condition-variant place recognition datasets reveals an average performance improvement of 13% over cutting-edge visual place algorithms (Khaliq et al., 2019). Their method stacks feature map activations to get local descriptors (Khaliq et al., 2019). The most prevalent areas must be determined to extract region-based convolutional neural network characteristics. Strong focus is placed on static features including buildings, trees, and road signs in convolutional-neural-network-based defined zones (Khaliq et al., 2019). They used 1125 photos of 365 locations shot at different times of day, night, and evening to teach a wide range of words (Khaliq et al., 2019). A test picture is compared to a reference image using the dot/scalar product of each region's VLAD components, after which the final single matching score is determined (Khaliq et al., 2019). Each test picture is then cosine-matched against each reference image, with the reference image with the greatest similarity score being chosen as the matched image in the end (Khaliq et al., 2019). Against R = 750 reference pictures, the overall retrieval times per query are between 446.405 and 533.245 ms (Khaliq et al., 2019).

The model suggested in this thesis is trained on scene-centric recognition, while it performs very well in visual place recognition (Olid et al., 2018). As Olid et al. approved that the model trained for scene recognition achieved better results in all the studied combinations (Olid et al., 2018). The primary explanation for this is as the internal layers of the CNN have learned to extract characteristics that are more advantageous for location recognition to classify scenes (Olid et al., 2018).

CHAPTER 2

2. CLUSTERING APPROACHES

In this study, we employ K-means to group and cluster the features of images obtained from AlexNet CNN response, SIFT, and HOG, utilizing Manhattan distance for measuring the similarity between the features.

2.1 Grouping Objects by Similarity Using K-Means

Finding groups of samples that are more related to one another than to samples in other groups is possible with the use of the approach known as clustering (or cluster analysis) (Raschka and Mirjalili, 2019). Grouping documents, music, and movies according to some criteria like subjects is an example of a business-oriented use for clustering (Raschka and Mirjalili, 2019). Another example is grouping consumers who have similar interests as the foundation for recommendation engines (Raschka and Mirjalili, 2019).

2.2 K-Means Clustering

The prototype-based clustering subcategory includes the k-means technique (Raschka and Mirjalili, 2019). With prototype-based clustering, each cluster is represented by a prototype, which is typically either the centroid (average) of similar points for continuous features or the medoid (the most representative or the point with the shortest distance to all other points that belong to a specific cluster) for categorical features (Raschka and Mirjalili, 2019). The k-means method may be used to accomplish our objective of grouping the samples based on their feature similarities, which can be summed up into the following four steps(Raschka and Mirjalili, 2019):

1. Choose k centroids at random.
2. Place each sample at the closest centroid, $\mu(j)$, $j \in \{1, \ldots, k\}$.

3. The centroids should be recalculated to be the middle (average) of all samples that belong to that class.
4. Once a user-defined tolerance or the maximum number of repetitions has been achieved process will stop, otherwise repeat steps 2 and 3 until the cluster assignments do not change anymore.

2.2.1 How Do We Measure Similarity Between Objects?

The squared Euclidean distance between two points, x and y, in m-dimensional space, is a frequently used measure for grouping instances with continuous data. The similarity is defined as the opposite of distance (Raschka and Mirjalili, 2019).

$$d(x,y)^2 = \sum_{j=1}^{m}(x_j - y_j)^2 = ||x - y||^2 \quad (1)$$

It's important to note that the index j in the previous equation relates to the jth dimension (feature column) of the sample inputs, x and y (Han et al., 2012).

Based on this Euclidean distance metric, we may characterize the k-means algorithm as a straightforward optimization problem, an iterative method for reducing the within-cluster sum of squared errors (SSE), also known as cluster inertia (Raschka and Mirjalili, 2019):

$$SSE = \sum_{i=1}^{n}\sum_{j=1}^{k} w^{(i,j)} \left|\left|x^{(i)} - \mu^{(j)}\right|\right|_2^2 \quad (2)$$

Here, $\mu^{(j)}$ is the representative point (centroid) for cluster j. $w^{(i,j)}$ = 1 if the example, $x^{(i)}$ is in cluster j, or 0 otherwise.

$$w^{(i,j)} = \begin{Bmatrix} 1\,, & if\ x^{(i)} \in j \\ 0\,, & therewise \end{Bmatrix} \quad (3)$$

In this work, we used the cluster module of Scikit-K-Means Learn's class.

2.3 AlexNet Layers

The AlexNet model comprises three fully-connected layers, three maximum pooling layers, five convolutional layers, and a 1000-way SoftMax classifier (Implementing AlexNet CNN Architecture Using TensorFlow).

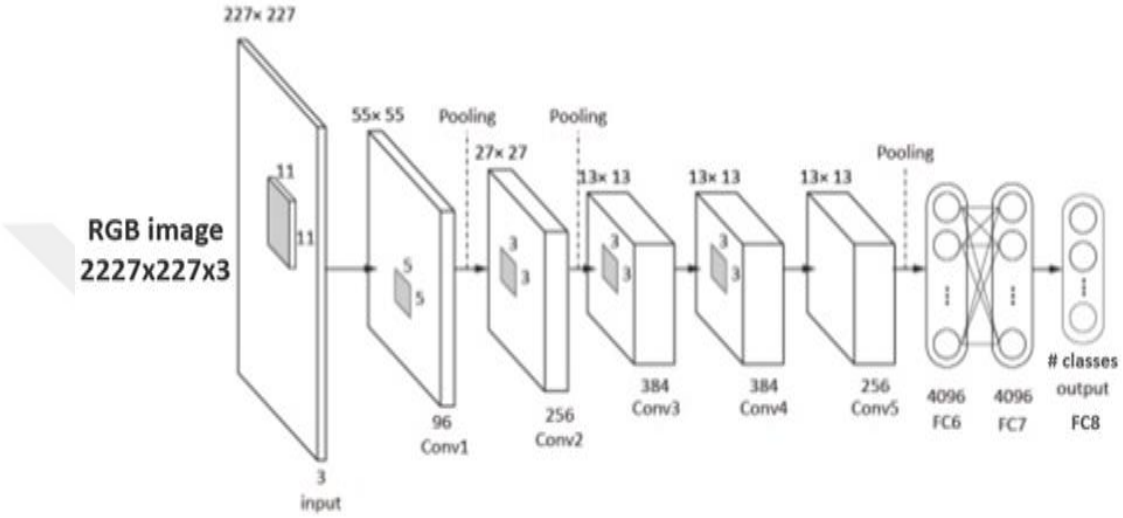

Figure 1: AlexNet Layers.

2.3.1 AlexNet Extract Image Features

Activations on the fully connected layer "fc7" are used to get the feature representations of the training and test pictures (AlexNet). Use a layer before the fully connected layer can be used to obtain a lower-level representation of the photos (AlexNet). A convolutional neural network's layers each create an activation in response to an input picture (Image Category Classification Using Deep Learning). However, only a handful of a convolutional neural network's layers are appropriate for extracting picture features (Image Category Classification Using Deep Learning).

When we apply layer = "conv3" to an input picture, we receive a response with the shape (13, 13, 384), which we then convert to a vector with the shape (64869). To obtain a vector that represents a picture after this procedure. A list of the image response vectors is created to represent an entire dataset of photos. Due to our use of the Garden Point Dataset. On the QUT campus, the Garden Point Dataset was recorded during two traverses: one during the day on the left sidewalk and the other during the night on the

right sidewalk (Khaliq et al., 2019). It is a test dataset with 200 photos taken in daylight from the left side of a sidewalk. A reference dataset also includes 200 photos taken at night and on the right sidewalk (Khaliq et al., 2019). As 200 photographs are feeding AlexNet in this experiment, we only use daytime images. The list of response vectors for the entire set of pictures with the form of (200, 64869). K-means clustering benefits from just accepting two-dimensional arrays, which is beneficial (X, Y).

2.3.2 Results Of AlexNet Image Features Classification

We must a priori select the number of clusters, k, to begin the K-means Cluster Algorithm. We choose k=25 for the number of clusters so the result of the distribution of images over cluster indices is shown in Figure 2, where every dot represents a cluster.

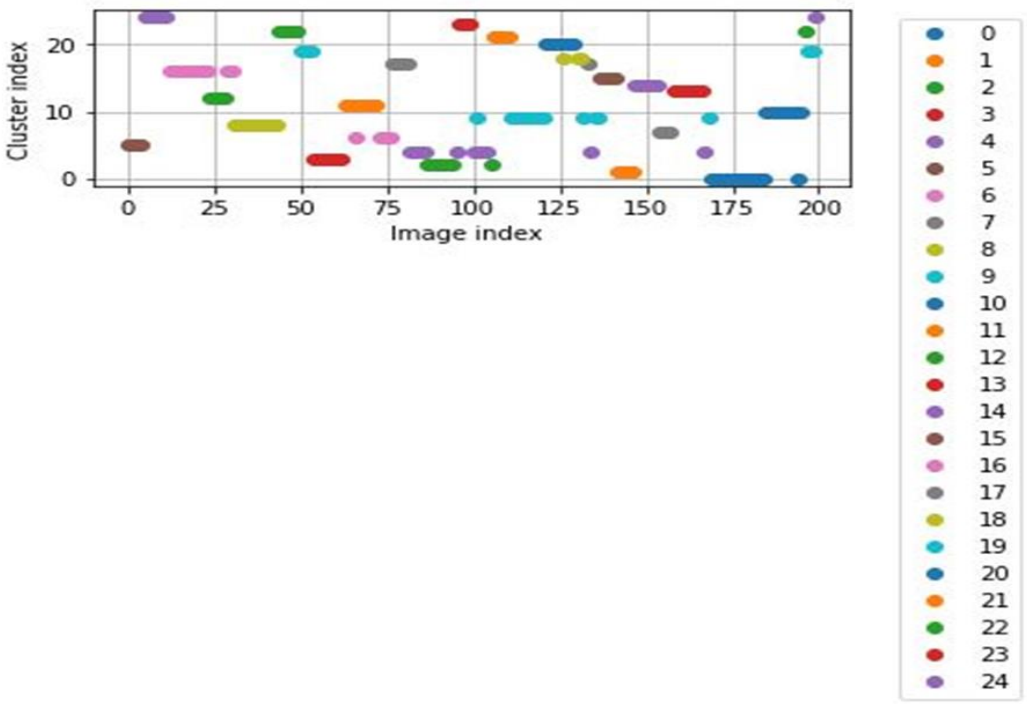

Figure 2: Images distribution over clusters.

Every dot in Figure 2 represents a cluster. We notice that nearly all images that are spatially close are grouped into the same cluster. The index of images ranges between 0 and 200. Despite being an unsupervised approach and not knowing the spatial distance between photos, the K-means clustering algorithm grouped images that were close to one another. Close photos are ones that are almost identical, thereby demonstrating that

AlexNet characteristics may be used to cluster images and that AlexNet perceives comparable images as being the same. Similar to cluster 1, which consists of the photos [142, 143, 144, 145, 146]. Cluster 1 contains pictures that are sequentially closer to one another. Figure 3 shows samples of the images that belong to cluster 7 and the nearest image to the center. Given, that the nearest image to the center of cluster 7 is 155 as shown in Figure 3.

Figure 3: Images belonging to cluster 7.

2.4 Using the Elbow Method to Find the Optimal Number of Clusters.

We must employ intrinsic measures, such as the within-cluster sum of squared errors (SSE), to compare the performance of various k-means clustering (distortion) (Raschka and Mirjalili, 2019). Figure 4 shows distortion and the number of clusters using K-means Clustering.

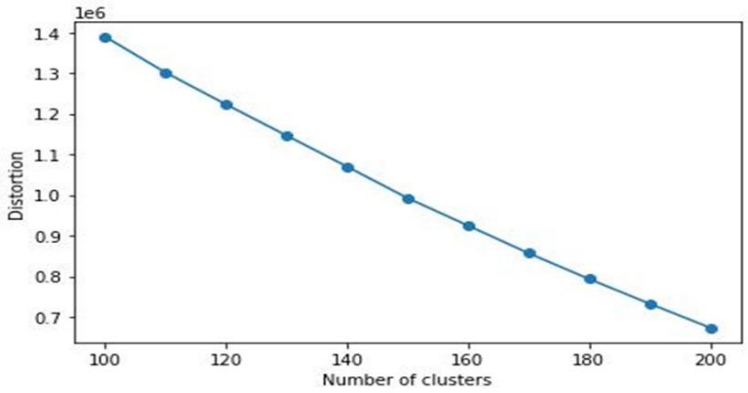

Figure 4: Distortion with the number of clusters.

As shown in Figure 4, when the number of clusters=100, the distortion is 1.4*10^6 and it has a slope till it reaches the number of clusters=200, then every image has its cluster and the distortion becomes 0.

2.5 How Time Does It Take To Cluster Images?

We investigated the number of clusters over time with the number of images 400 on Colab (Without GPU or TCP configuration), and the result is shown in Figure 5.

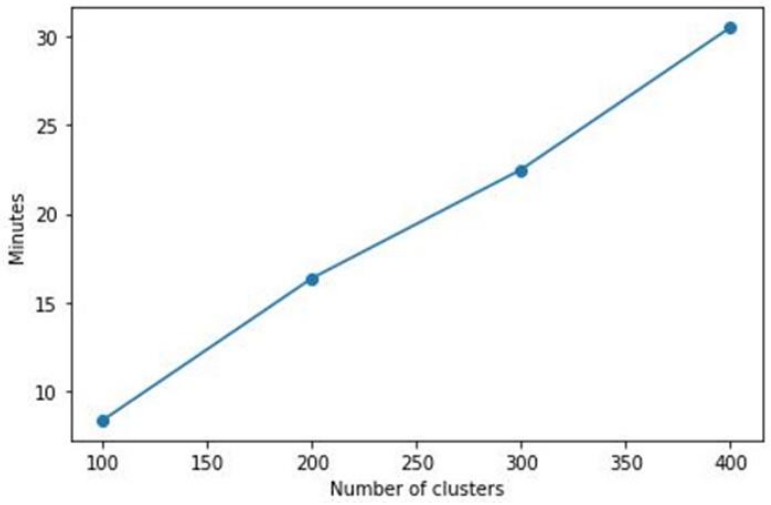

Figure 5: Time vs the number of clusters.

Figure 5 shows when the number of clusters is 100 it takes 08:33 minutes to classify images, when the number of clusters becomes 400 it takes 30:54 minutes to classify them.

2.6 Quantifying the Quality of Clustering Via Silhouette Plots

Silhouette analysis is another inherent metric to assess the quality of a grouping (Raschka and Mirjalili, 2019). It can be used as a graphical tool to visualize how closely the samples in the clusters are clustered (Han et al., 2012). We may use the following three stages to get the silhouette coefficient for each sample in the used dataset (Han et al., 2012):

1. The average distance between a sample point, $x^{(i)}$ and every other point in the same cluster is used to calculate the cluster cohesiveness, or $a^{(i)}$ (Han et al., 2012).

2. As the average distance between the example, $x^{(i)}$ and all the instances in the closest cluster, determine the cluster separation, $b^{(i)}$ from that cluster (Han et al., 2012).

3. To calculate the silhouette, $s^{(i)}$ divide the difference between cluster cohesiveness and separation by whichever is bigger, as illustrated below (Han et al., 2012) in Equation (4):

$$s^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max\{b^{(i)}, a^{(i)}\}} \quad (4)$$

The silhouette coefficient has a limit between -1 and 1. The silhouette coefficient is 0 if the cluster separation and cohesion are equal $(b^{(i)} = a^{(i)})$, according to the equation above (Raschka and Mirjalili, 2019). Since $b^{(i)}$ quantifies how different an example is from other clusters and $a^{(i)}$ informs us how similar it is to the other instances in its own cluster, we approach close to an ideal silhouette coefficient of 1 if these two parameters are equal (Han et al., 2012). Figure 6 shows a silhouette plot, given that the optimal number of clusters was found to be 65.

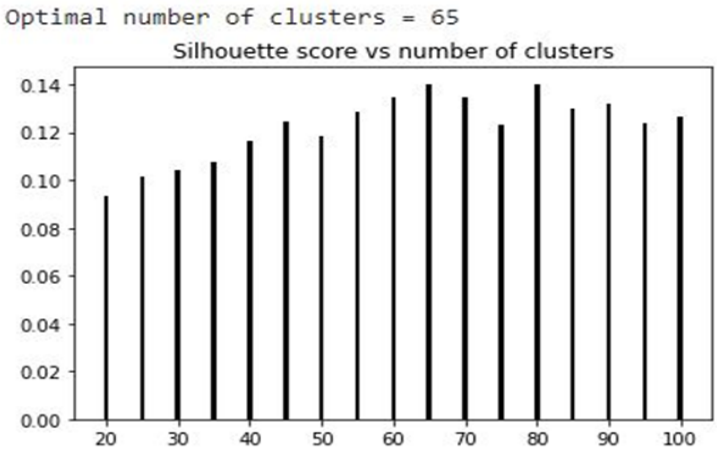

Figure 6: Silhouette with the number of clusters.

2.7 Manhattan Distance

The total of the lengths of the projections of the line segment between the points onto the coordinate axes determines the Manhattan distance, or d_1, between two vectors p, q in an n-dimensional real vector space with a fixed Cartesian coordinate system (Taxicab geometry).

$$d_1(p,q) = |p \quad - q \quad|^1 = \sum_{i=1}^{n} |p_i - q_i| \qquad (5)$$

Where (p, q) are the vectors of image features, p=(P_1, P_2,….., P_n) and q=(q_1, q_2,….., q_n). For example, in the plane, the Manhattan distance between (P_1, P_2) and (q_1, q_2) (Taxicab geometry) is:

$$d_1(p,q) = |p_1 - q_1| + |p_2 - q_2| \qquad (6)$$

2.7.1 Classify Images Using Manhattan Distance

We propose a method for categorizing images based on the Manhattan distance, where the image x_i serves as the focal point of the cluster C_i and all images with a distance of less than or equal to d are grouped together. Where i=1, 2,..., 200, and d is the categorization threshold that we set through trial. We categorize the photos in an AlexNet

response using Threshold=3150 after applying the Manhattan distance. Figure 7 displays the results of the Manhattan Distance to AlexNet Response.

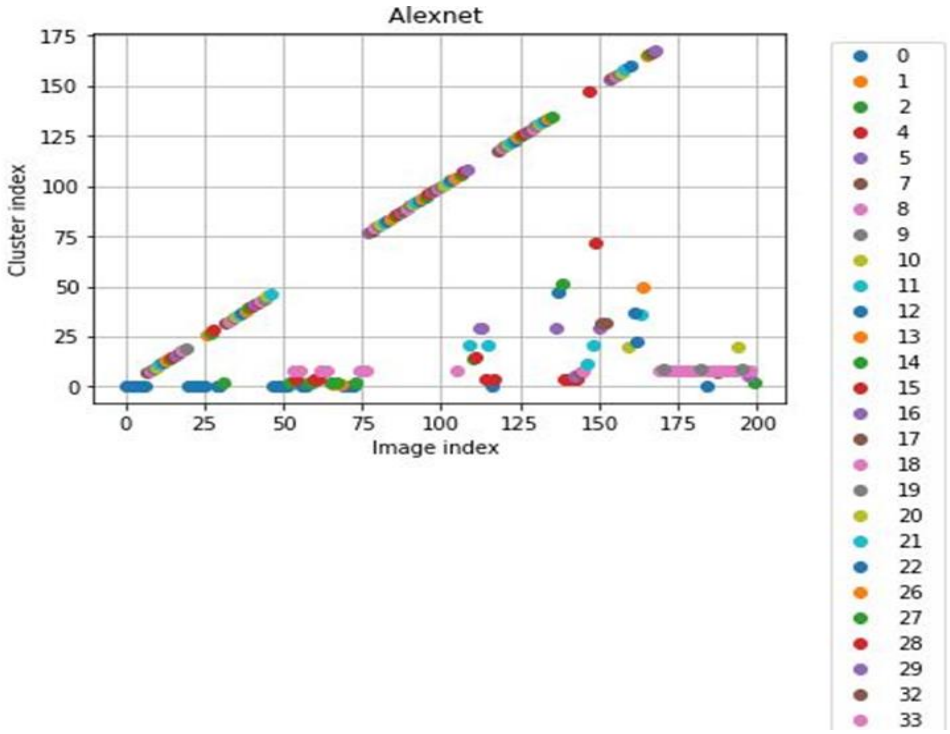

Figure 7: Clustering Using Manhattan Distance to AlexNet Response.

Every dot in Figure 7 represents a cluster. Figure 7 demonstrates how some photographs are grouped together, while others as indicated by the straight line, only include one image per cluster.

The Manhattan Distance clusters far-removed photos together, even though it is a distance measurement and cannot determine the spatial closeness between images. Manhattan Distance doesn't group nearby photos together since it assumes that far images are almost non-similar, even if characteristics from AlexNet are a useful tool for image clustering. The set of photos in cluster 7 ([7, 144, 187]). Cluster 7 has no consecutively closer photos to one another as shown in Figure 8. The Manhattan Distance is a poor technique for clustering.

Figure 8: Images belonging to cluster 7.

2.8 SIFT - Scale-Invariant Feature Transform

To find and characterize local features in digital pictures, one approach is used: the scale-invariant feature transform (SIFT) (SIFT-Scale-Invariant Feature Transform). It finds specific focal locations and provides them with quantitative data (known as descriptors), which may be applied, for instance, to object recognition. The descriptors should be invariant to various modifications that might cause photos to seem different even though they depict the same item.

2.8.1 SIFT Feature Descriptor Generation

The SIFT algorithm's final step is to produce the descriptor, which is a normalized 128-dimensional vector. At this phase in the process, a list of feature points that are specified in terms of their position, size, and orientation is given to us by SIFT. This makes it possible for us to create a local coordinate system centred on the feature point that should be consistent between multiple views of the same feature.

Every key point in SIFT features has a vector of 128 dimensions; key points in images vary; some images have more key points than others; however, in K-means algorithms, we need a fixed number of key points for the entire set of images. We decided that every image should have 128 key points, and any image with fewer than that number is complemented with 0s. For instance, if a picture has 110 key points, we would add 0s to rows 111 through 128 to ensure that all of the photos in the list have the same structure

(128,128). As a result, the dataset features now have the shape (200,128 * 128), where 200 is the number of images in the dataset, 128 are key points, and 128 is a SIFT descriptor. This array contains the entire SIFT features of images as an input to K-means Clustering. Figure 9 displays K-means Clustering results and a list of SIFT Features of input photos with a total of 25 clusters:

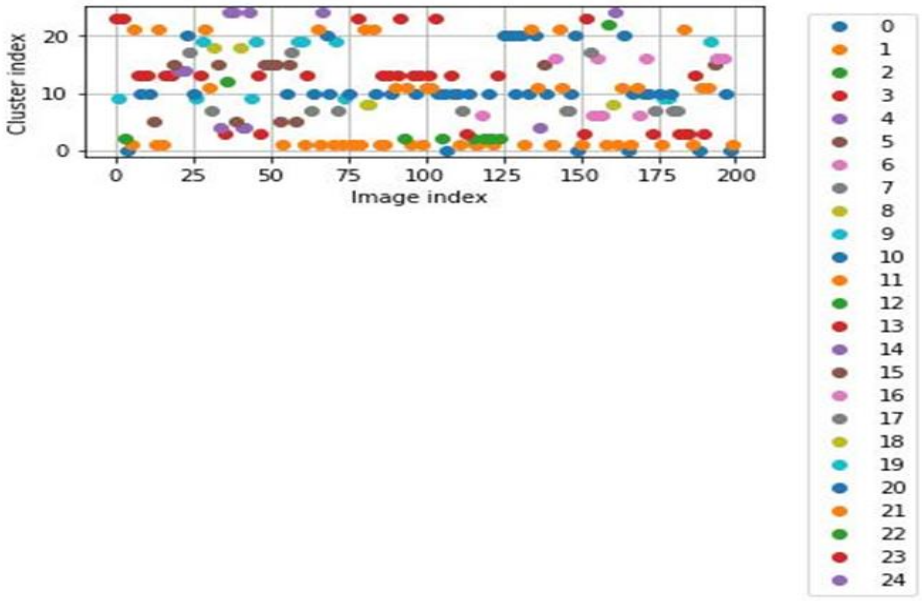

Figure 9: Results of K-Means Clustering and a List of SIFT Features.

Figure 9 shows images that are not spatially close to each other are clustered together, and the SIFT algorithm cannot group images spatially close as AlexNet did.

To get the average distance between a given example, $x^{(i)}$, and all other points in the same cluster, use the formula $a^{(i)}$, as shown in (Raschka and Mirjalili,2019). The silhouette plot displays negative values, indicating that the average distance within a cluster is greater than the average distance between clusters. Hence, Figure 10 highlights the SIFT feature's bad clustering intuition.

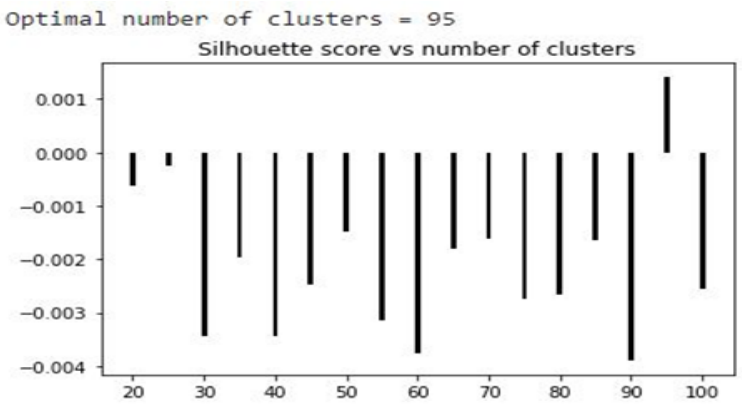

Figure 10: Silhouette Plot of Clustering Using SIFT Features.

The K-means clustering algorithm, however, is an unsupervised method and it is unaware of the geographical separation of the pictures. SIFT features are a poor technique for picture clustering; when combined with K-means, SIFT features provided subpar results. For example, cluster 5 contains pictures that are not sequentially closer to one another as shown in Figure 11. This demonstrates why SIFT features are a poor clustering technique.

Figure 11: Images belonging to cluster 5.

2.8.2 Manhattan Distance with SIFT Features.

135 clusters are present when Threshold=1800 is used, while the number of clusters is five when the threshold=2000. The outcome of threshold=1895 is shown in Figure 12, where Manhattan distance is applied to SIFT features of images which results in having 98 clusters.

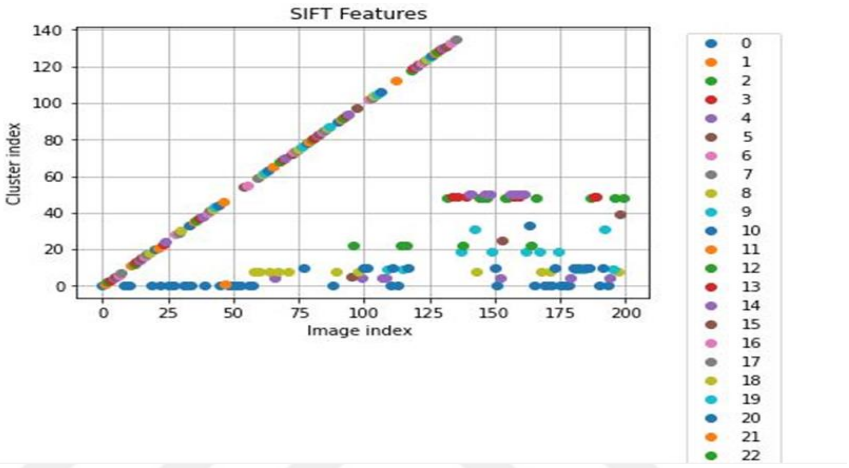

Figure 12: Clustering using Manhattan Distance and SIFT Features.

Figure 12 also shows some images are clustered together even though they are not spatially close together, which means SIFT feature algorithms are inappropriate for spatial clustering.

The Manhattan Distance clusters far-removed images together, even though it is a distance measurement and cannot determine the spatial closeness between images. Far images imply almost unrelated ones, SIFT characteristics are poor tools for image clustering, and Manhattan Distance doesn't group near images together. This demonstrates that Manhattan Distance and SIFT features alone are poor clustering methods. Similar to cluster 4, which contains the following images: [4, 66, 99, 107, 108,

152, 179, 194]. Cluster 4 contains images that are not sequentially closer to one another as shown in Figure 13.

Figure 13: Images belonging to cluster 4.

2.9 Histogram of Oriented Gradients

A feature descriptor called HOG, or Histogram of Oriented Gradients, is frequently employed to extract features from picture data (Feature Engineering for Images). It is commonly used for object identification in computer vision tasks. Let's examine a few key features of HOG that set it apart from other feature descriptors:

- The HOG description emphasizes an object's structure or form. You might be wondering how this differs from the edge characteristics we extract for photos at this point (Feature Engineering for Images). We only determine if a pixel is an edge in the case of edge features. HOG is also able to offer the edge direction. This is accomplished by separating the gradient and orientation (or magnitude and direction, depending on your preference) of the edges.
- These orientations are also computed in "localized". This implies that the entire image is divided into smaller sections, and the gradients and orientation are determined for each region.

- A Histogram would then be produced by the HOG for each of these areas independently. The term "Histogram of Oriented Gradients" refers to the histograms that are produced using the gradients and pixel values that are oriented in a certain direction.

To give this a formal definition:

The HOG feature descriptor keeps track of the instances of gradient orientation in certain areas of an image.

Every picture has a HOG descriptor that is a vector of shape (3780), hence an array of shapes (200, 3780) may be constructed from a list of photos, where 200 is the number of images in the dataset. The list of photos HOG will be a K-means Clustering input. Figure 14 displays the outcomes of using K-means with 25 clusters.

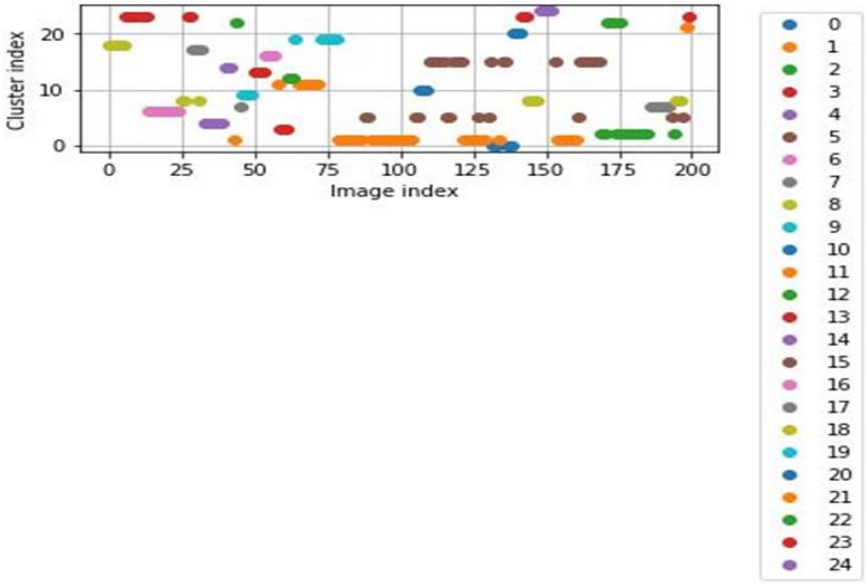

Figure 14: HOG with K-means Clustering.

Figure 14 shows nearly spatially close images are classified together and give better results than SIFT. As mentioned before, K-means clustering ignores the spatial relationship between the pictures. However, nearly all closed pictures were clustered together by combining HOG characteristics with K-means. Accordingly, demonstrating

characteristics collected by HOG is a useful method for clustering photos since close images represent roughly comparable images. The collection of photos in cluster 4 ([33, 34, 35, 36, 37, 38, 39]). In cluster 4, the photos are arranged in a more orderly fashion as shown in Figure 15.

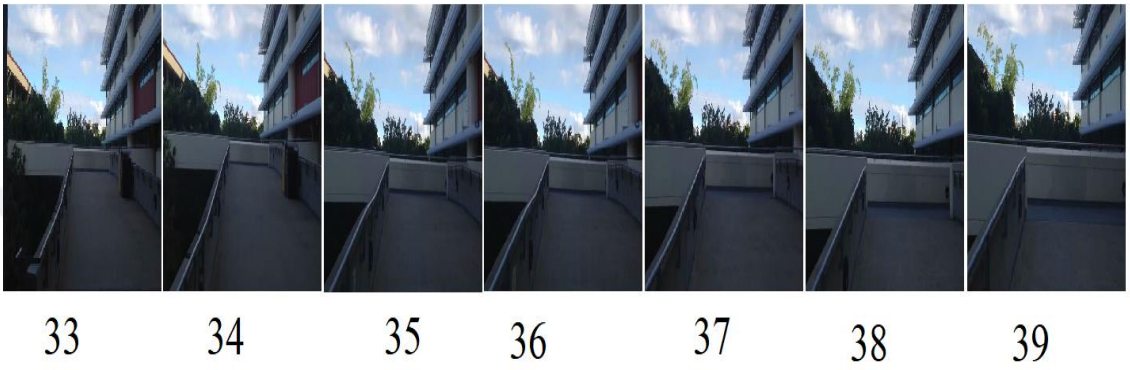

Figure 15: Images belonging to cluster 4.

We can demonstrate that by silhouette plot, as shown in Figure 16, a positive result near 0.08 which means more cohesion inside the cluster:

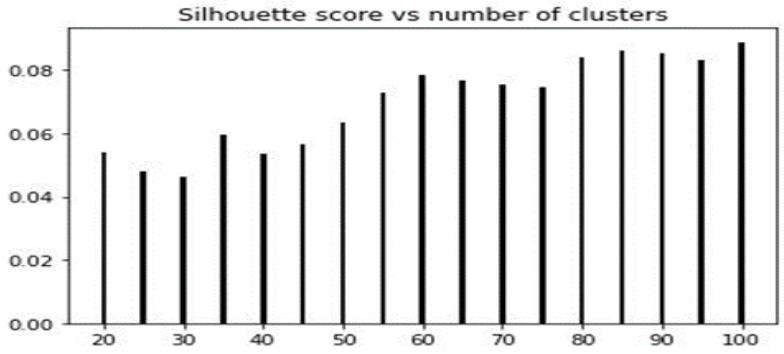

Figure 16: Silhouette for Clustering Using HOG.

2.9.1 Manhattan Distance with HOG Features.

118 clusters are present when Threshold=340 is used. The amount of clusters is 88 if Threshold=370. The outcome of threshold=355 is shown in figure 17 when we apply Manhattan distance to HOG features of pictures with 107 clusters.

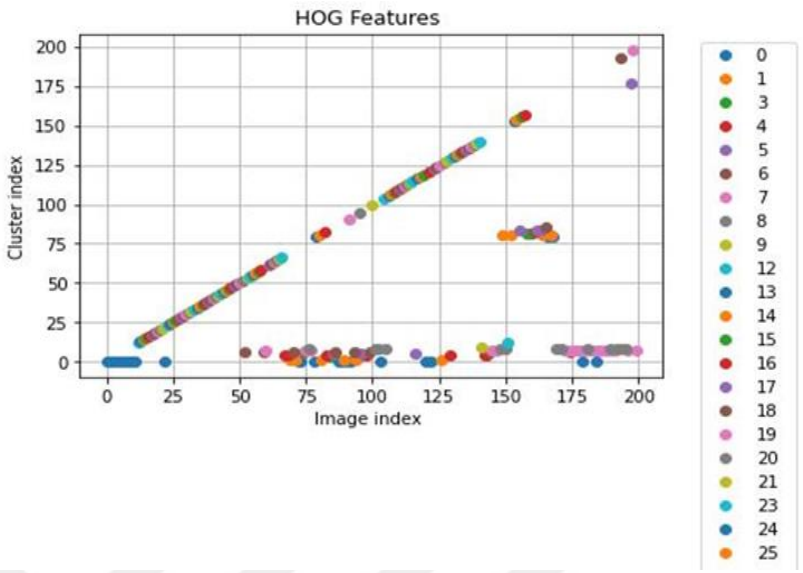

Figure 17: Clustering using Manhattan Distance and HOG Features.

Figure 17 demonstrates how some photographs are grouped together while others, as indicated by the straight line, only include one image per cluster. Although the Manhattan Distance is a measure of distance and does not take into account the spatial proximity of pictures, it grouped those with a great spatial distance apart. While HOG characteristics are effective tools for image clustering and Manhattan Distance doesn't cluster close photos to one other, far images signify images that are almost unrelated to one another. Therefore, the Manhattan Distance and HOG features are ineffective clustering methods when combined. The photos in cluster 6 ([52, 59, 70, 74, 86, 93, 99, 144, 174]) are clustered together. There is no sequential proximity between the photos in cluster 6 of the cluster as shown in Figure 18.

Figure 18: Images belonging to cluster 6.

2.10 Comparing Clustering Method using AlexNet, HOG and SIFT Features

The results from the AlexNet response are better than those from HOG, as shown by the findings of the three methodologies we tested—SIFT features, HOG features, and AlexNet response. Our findings for spatial classification from SIFT are below average. The response time for AlexNet is however longer. While HOG yields result rapidly, it is nevertheless effective.

CHAPTER 3

3. COMPARİNG BETWEEN ALEXNET AND RESNET 18

Both ResNet 18 and AlexNet are thoroughly examined in this section. We have investigated the official version as well as the effect of removing some layers on both accuracy and reaction time. Here, we first demonstrated how to form clusters, then we displayed the original structure clusters. A better model will be selected as a primary block for this research based on the results of this investigation. A portion of the St. Lucia dataset (Warren et al., 2010) was utilized. This is the first part of St. Lucia (Warren et al., 2010) consists of 1688 images. Other parts right now (2022) are not available on the website of St. Lucia (Warren et al., 2010).

3.1 Clustering.

Utilizing cosine similarity as a clustering Method. Cosine similarity can be expressed in Equation (7):

$$Similairty\ (A,B) = \frac{A.B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (7)$$

When calculating the cosine similarity between the HOG of two images, we used a threshold equal to 70% to cluster images together. In other words, if

1. Cosine similarity >=70%: the compared images belong to the same cluster.
2. Cosine similarity<70%: then they belong to different clusters.

The cosine similarity was set to 70% based on investigating multiple values and the details of the investigated values are shown in table 12.

3.2 Clustering and Labeling.

The Clustering method produces the labelling, where all Images in the same cluster have the same label. For instance, photos from [1-286] are part of the first cluster and receive the label "0". Similarly, the last cluster includes the images from 1603 to 1688,

which is why they receive "124" labels. The issue here is that certain classes contain around 288 photos, whereas some other classes only contain two photos.

3.3 Creating Dataset

We first scaled down all the images in the St. Lucia Dataset to (227,227), so they can be used with Places365 Networks. Next, images are transformed from Bayer mode to BGR and then back to RGB before being stored. As a result, there are 125 clusters and 1688 photos in total. Then, each picture was transformed into a tensor and normalized using the ImageNet Method, Equation (8) is the predefined vectors used for the normalization process:

$$\text{imagenet\_stats} = ([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]) \quad (8)$$

Next, we split the dataset as follows:

1. The training set is 80%.
2. The validation set is 10%.
3. The test set is 10%.

Note that, using batch size 32. Now, each picture has a label and a form [3, 227, 227].

3.4 AlexNet Architecture

Krizhevsky et al. proposed AlexNet in 2012. They recommended using AlexNet to categorize the 1.2 million high-resolution photos entered in the ImageNet LSVRC-2010 competition into 1000 separate groups (Krizhevsky et al., 2017). Their results were better than the prior state-of-the-art (Krizhevsky et al., 2017). AlexNet consists of three fully connected layers, five convolutional layers, some of which are followed by max-pooling layers, and a final 1000-way softmax layer (Krizhevsky et al., 2017). It includes 60 million parameters and 650,000 neurons (Krizhevsky et al., 2017). Additionally, AlexNet Places365 has been trained on 1.8 million photos from 365 different scene categories, where there are only a maximum of 5000 images per category, making it ideal for place recognition (Prykhodchenko and Skruch, 2022). Figure 19 shows the architecture of AlexNet.

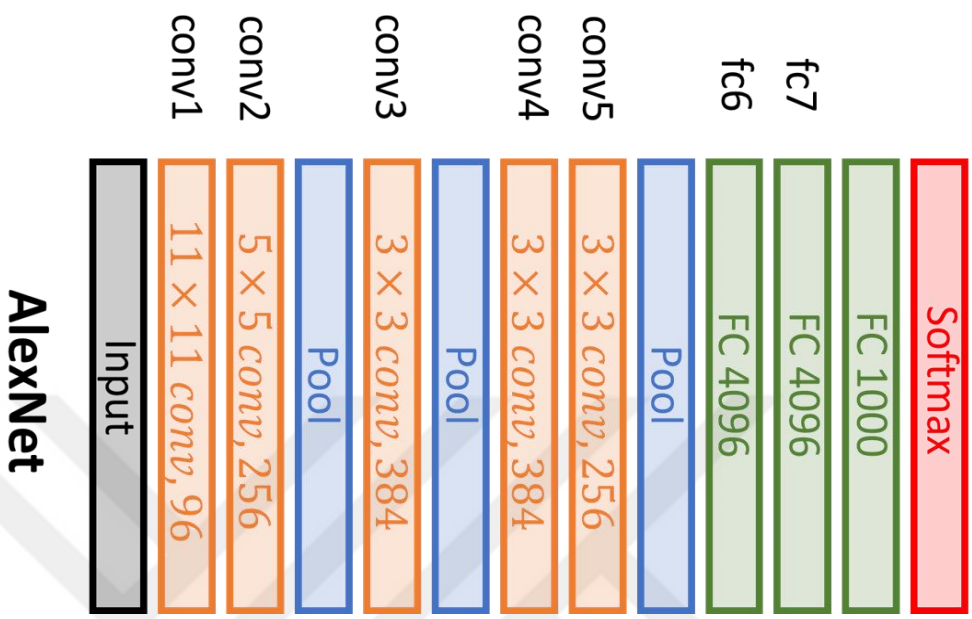

Figure 19: AlexNet Architecture.

We used the first part of the St.Lucia (Warren et al., 2010) dataset to compare the performance, accuracy, size and response time of ALexNet and ResNet18. This part consists of 1688 images.

3.4.1 AlexNet setup

Replace layer 6, the last fully connected layer that generates outputs of 365 (the number of categories in Place365), with a different fully connected layer that generates outputs of 125 (the number of classes in our dataset). In this work, we have completely frozen the layers except the last fully linked layer is being trained.

Figure 20 shows the accuracy of AlexNet with 40 epochs. Note that the accuracy drops dramatically after 4 epochs.

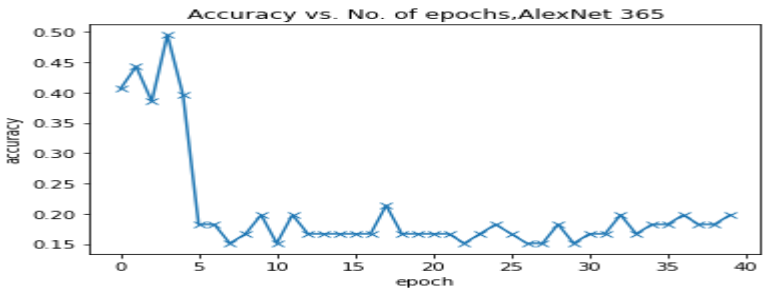

Figure 20: Accuracy of AlexNet with 40 epochs.

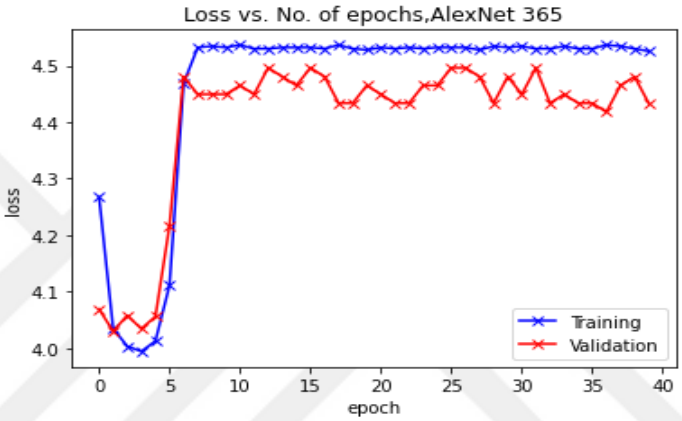

Figure 21: Loss of AlexNet with 40 epochs.

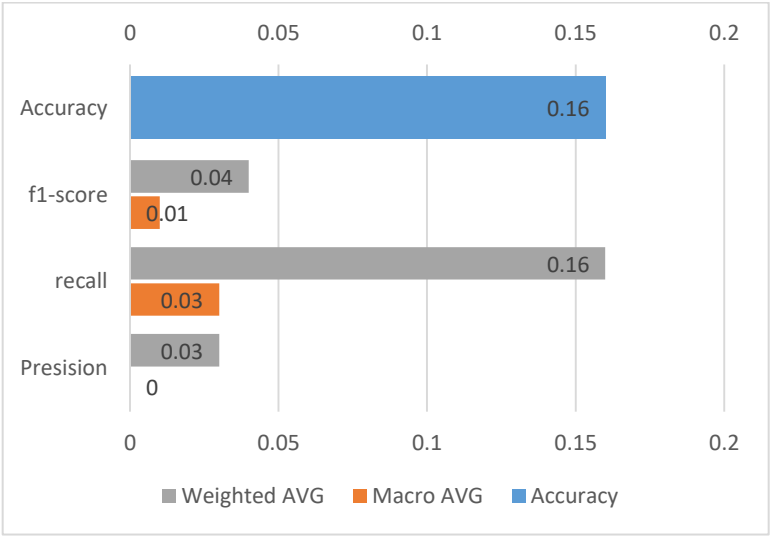

Figure 22: AlexNet results.

Table 1: AlexNet Parameter Information.

Total params	**57,524,159**
Trainable params	**57,524,159**
Non-Trainable params	**0**
Input Size (MB)	**0.59**
Forward/backward pass Size	**8.48**
Params Size (MB)	**219.44**
Estimated Total Size (MB)	**228.51**

AlexNet here displayed a forgetting behaviour, showing 50% accuracy in the first 4 epochs, then dropping to 20% in the fifth epoch. Two explanations for that inappropriate behaviour:

1. AlexNet uses a shallow architecture.
2. As the used dataset has unbalanced classes that might lead the network to incorrectly categorize certain classes while remembering the larger classes.

Now we will remove the last two layers of AlexNet and display accuracy as Figure 23 shows.

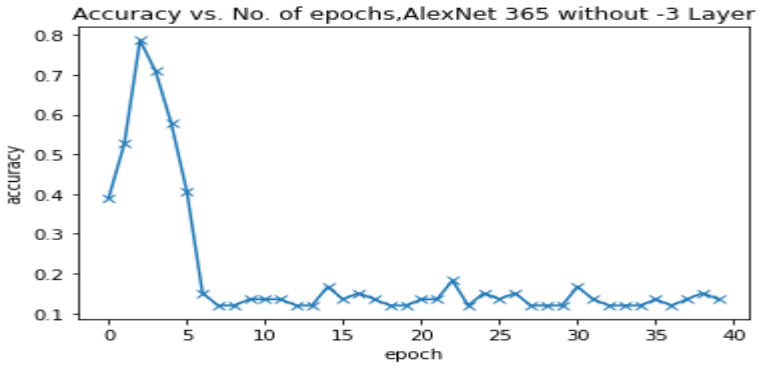

Figure 23: Accuracy AlexNet without 2 last layers.

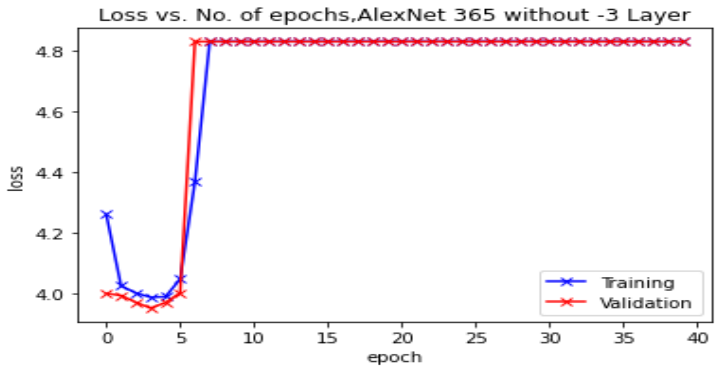

Figure 24: Loss of AlexNet without 2 last layers.

Figure 23 shows the accuracy of AlexNet with 40 epochs, it showed accuracy drops dramatically after 5 epochs.

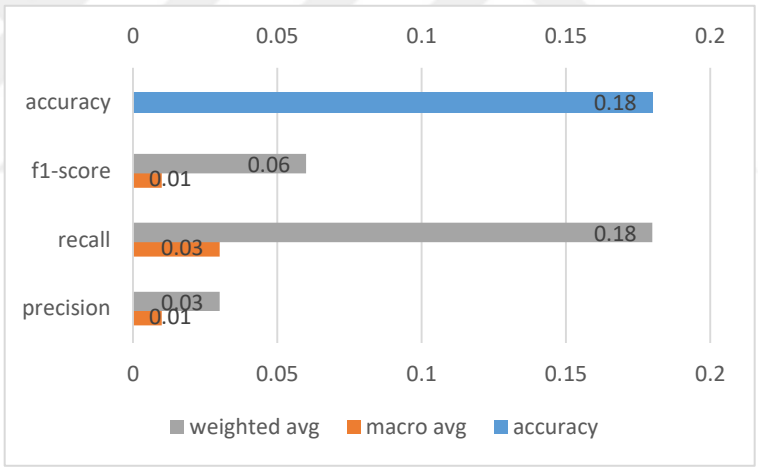

Figure 25: Results of AlexNet without the Last Two Layers.

Table 2: AlexNet Parameter Information without the Last Two Layers.

Total params	40,742,847
Trainable params	40,742,847
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	8.39
Params Size (MB)	155.42
Estimated Total Size (MB)	164.4

Now we will remove the last three layers of AlexNet and display accuracy as Figure 26 shows.

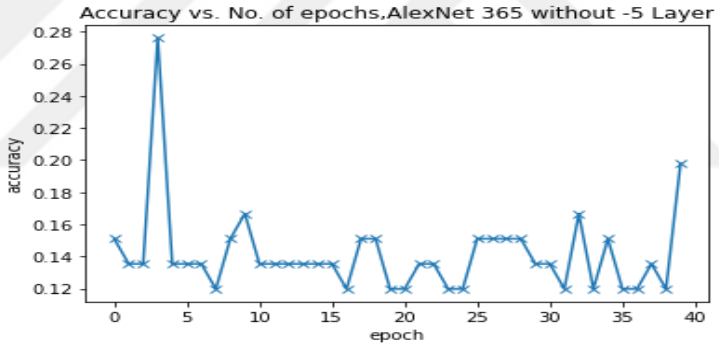

Figure 26: Accuracy of AlexNet without 3 last layers.

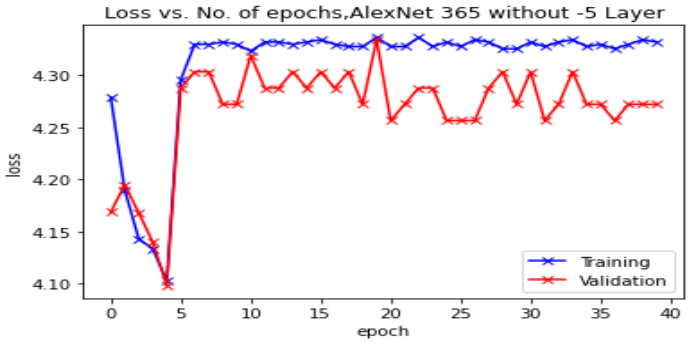

Figure 27: Loss of AlexNet without the last 3 layers.

Figure 26 shows the accuracy of AlexNet with 40 epochs, it shows accuracy just reaches 28% then drops dramatically after 4 epochs.

Table 3: AlexNet Parameter Information without the last three layers.

Total params	40,742,847
Trainable params	40,742,847
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	8.36
Params Size (MB)	155.42
Estimated Total Size (MB)	164.37

Table 4: AlexNet without last layers Accuracy, average precision, size and average response time in milliseconds.

Model	**ACC**	**AVG P**	**Size**	**Response Time in a millisecond**
AlexNet	0.5	0.07	228 MB	46 ms
AlexNet-2	0.8	0.01	164 MB	43 ms
AlexNet-3	0.28	0.06	164 MB	40 ms
AlexNet without 6 last layers	0.58	0.04	23 MB	33 ms

Table 4 shows AlexNet without the last two layers is better than the original AlexNet and AlexNet with the last three layers. But in general, AlexNet here shows forgotten behaviour with increased epochs.

3.5 ResNet18 Architecture

He et al. provided a residual learning framework to facilitate the training of networks that are far deeper than those previously employed (He et al., 2016). They specifically reformulate the layers so that they learn residual functions with reference to the inputs of the layer rather than learning unreferenced functions (He et al., 2016). They give detailed empirical evidence demonstrating that these residual networks are simpler to tune and can significantly boost accuracy (He et al., 2016). They earned first place in the ILSVRC 2015 classification task. Additionally, they took first place in the tasks of ImageNet localization, COCO detection, and segmentation (He et al., 2016). They implement a deep residual learning approach to overcome the deterioration issue (He et al., 2016). They deliberately allowed these layers to suit a residual mapping rather than trusting that each of the few stacked levels would match a desired underlying mapping (He et al.,2016). ResNet Places365 is effective for place recognition since it has been trained on 1.8 million photos from 365 scene categories, where there are only at most 5000 images per category (Prykhodchenko and Skruch, 2022).

Figure 28 shows the architecture of ResNet 18, layers and residual blocks.

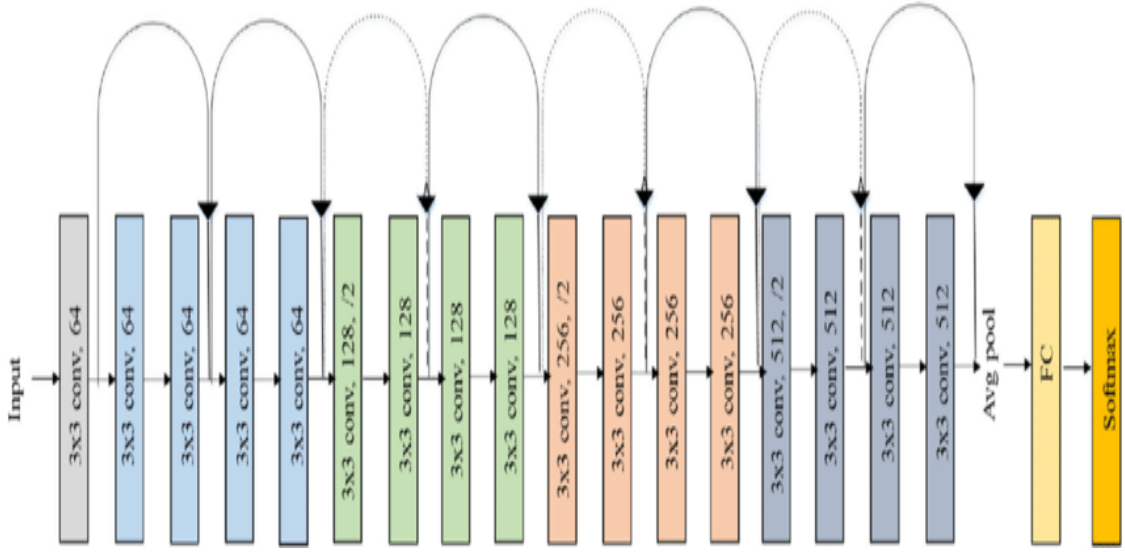

Figure 28: ResNet 18 Architecture.

3.5.1 ResNet 18 Setup

ResNet-18 has 18 layers and in accordance with the number of classes in our dataset, we replaced the last fully connected layer with another fully connected layer that has 125 output neurons. Right now, just the final completely linked layer is being trained after freezing all of the prior layers.

Here also we evaluate ResNet 18 on part of the St.Lucia dataset (Warren et al.,2010) to compare performance, accuracy, size and response time. This part consists of 1688 images.

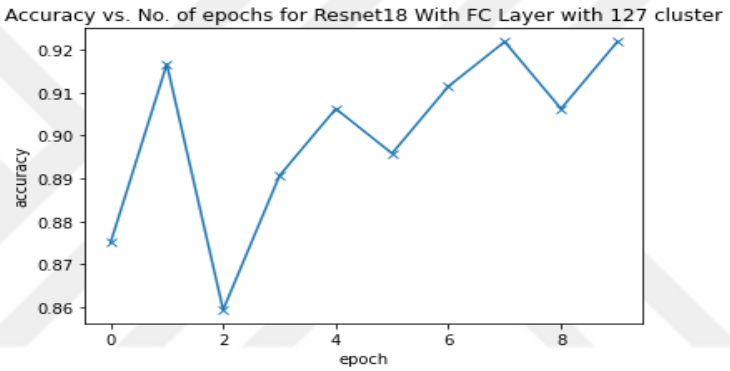

Figure 29: Accuracy of ResNet 18 with 10 epochs.

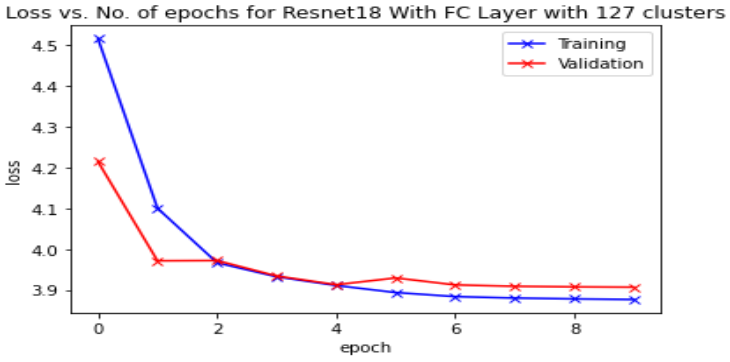

Figure 30: Loss of ResNet 18 with 10 epochs.

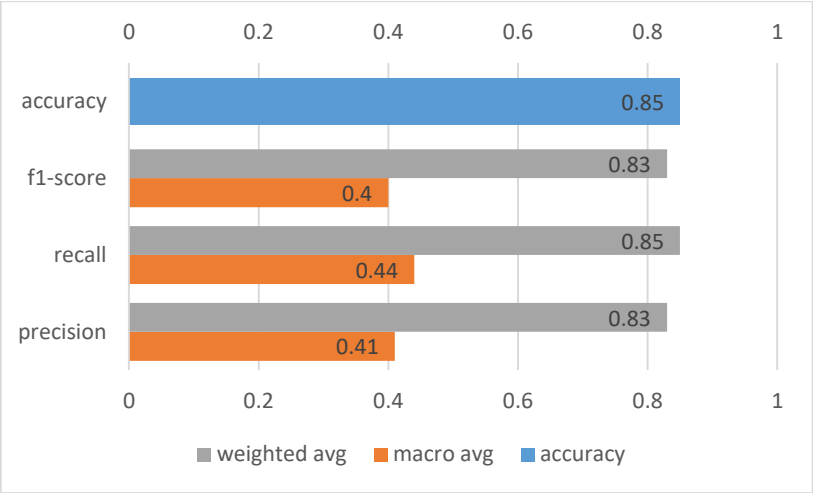

Figure 31: Results of ResNet 18.

Table 5: ResNet 18 Parameter Information.

Total params	11,240,637
Trainable params	11,240,637
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	67.01
Params Size (MB)	42.88
Estimated Total Size (MB)	110.48

Now we will remove the last layer of ResNet 18 and display accuracy as shown in Figure 32. It shows unstable behaviour, as it increased in the first 4 epochs, then it dropped dramatically.

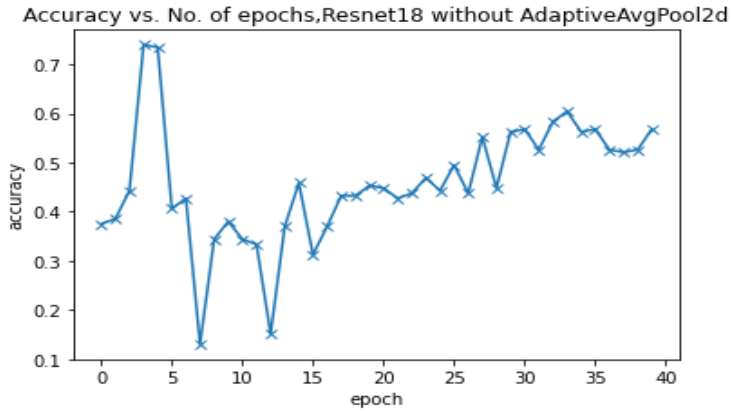

Figure 32: Accuracy of ResNet 18 without last layers with 40 epochs.

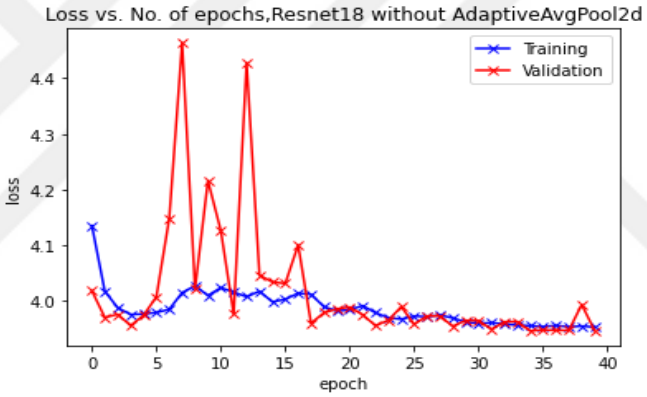

Figure 33: Loss of ResNet 18 without last layers with 40 epochs.

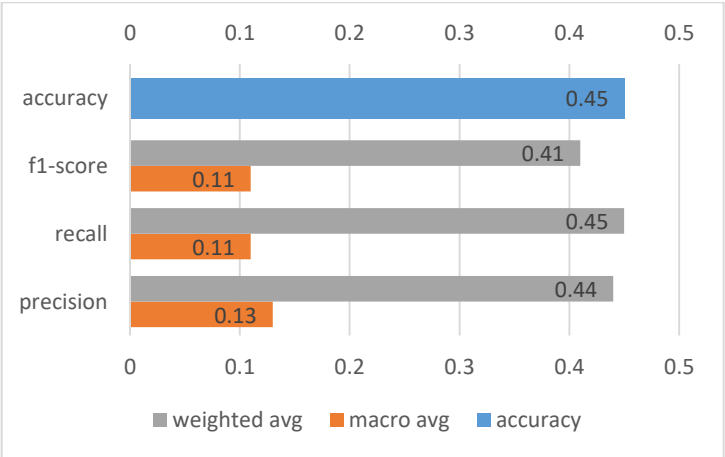

Figure 34: Results of ResNet 18 without the last layers.

Table 6: ResNet 18 Parameter Information without last layers.

Total params	15,272,637
Trainable params	15,272,637
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	67.26
Params Size (MB)	58.26
Estimated Total Size (MB)	126.11

Now we will remove the last two layers of ResNet 18 and display accuracy. As Figure 35 shows, good behaviour was achieved, as it increased after 2 epochs and still increased till it reached nearly 100%.

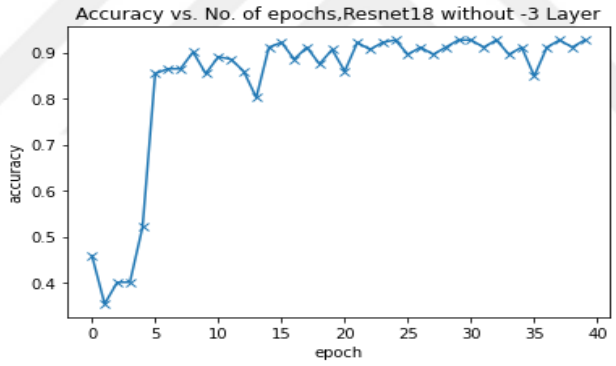

Figure 35: Accuracy of ResNet 18 without the last two layers with 40 epochs.

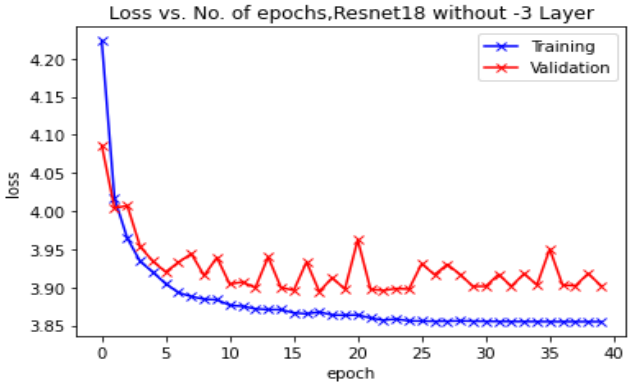

Figure 36: Loss of ResNet 18 without the last two layers with 40 epochs.

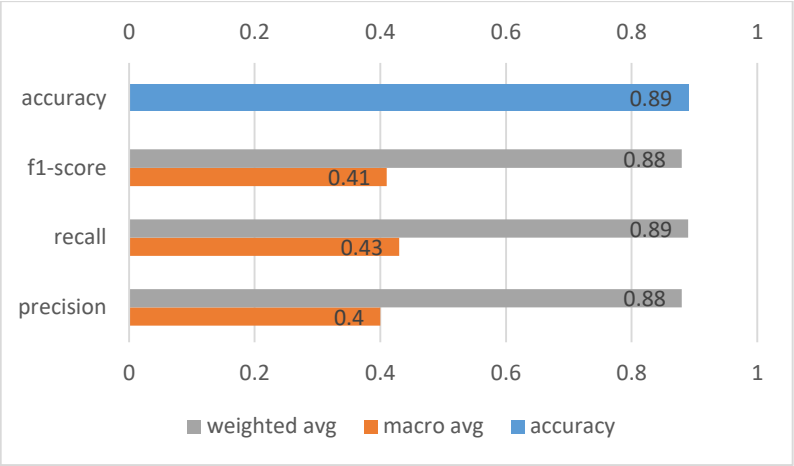

Figure 37: Results of ResNet 18 without the last two layers.

Table 7: ResNet 18 Parameter Information without the last two layers.

Total params	9,982,909
Trainable params	9,982,909
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	63.45
Params Size (MB)	38.08
Estimated Total Size (MB)	102.12

Now we will remove the last three layers of ResNet 18 and display accuracy as Figure 38 shows. It shows unstable behaviour, as it increased and dropped after every 2 epochs, but in general, still increased till it reached nearly 45%, then it started dropping.

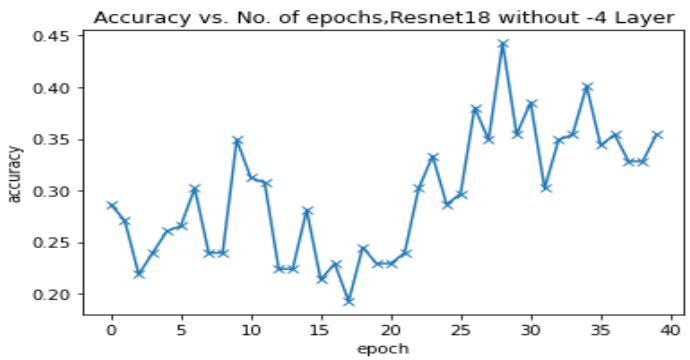

Figure 38: Accuracy of ResNet 18 without the last three layers with 40 epochs.

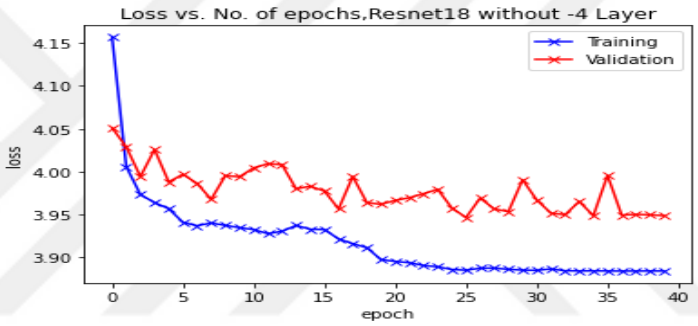

Figure 39: Loss of ResNet 18 without the last three layers with 40 epochs.

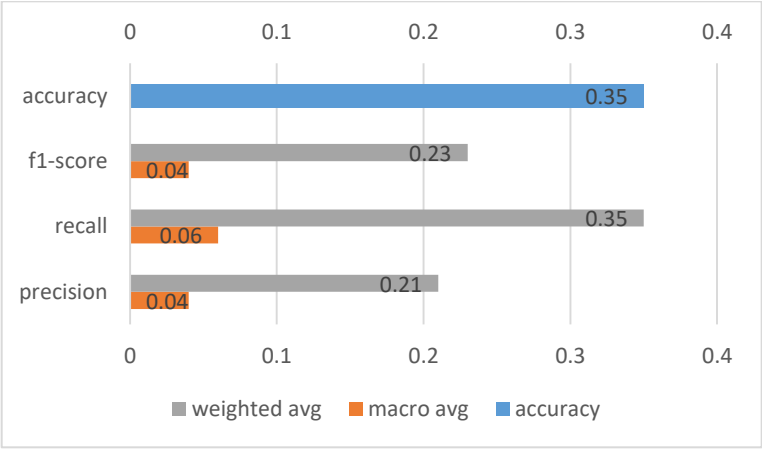

Figure 40: Results of ResNet 18 without the last three layers.

Table 8: ResNet 18 Parameter Information without the last three layers.

Total params	14,139,197
Trainable params	14,139,197
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	56.80
Params Size (MB)	53.94
Estimated Total Size (MB)	111.32

Now we will remove the last four layers of ResNet 18 and display accuracy as Figure 41 shows. It shows unstable behaviour, as it increased and dropped after every 2 epochs, with maximum accuracy reaching 20%.

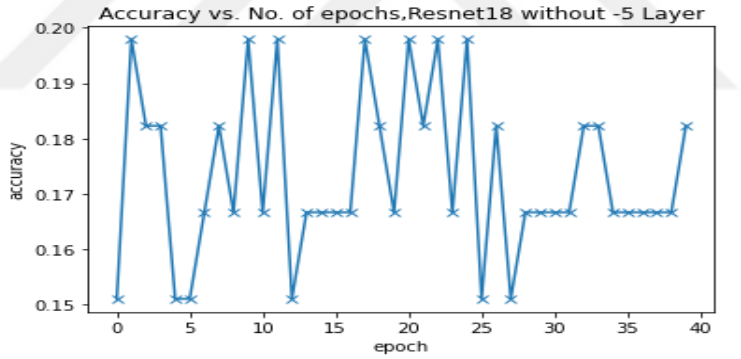

Figure 41: Accuracy of ResNet 18 without the last four layers with 40 epochs.

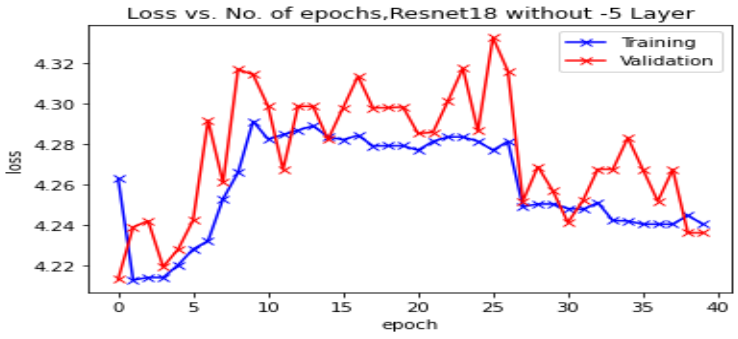

Figure 42: Loss of ResNet 18 without the last four layers with 40 epochs.

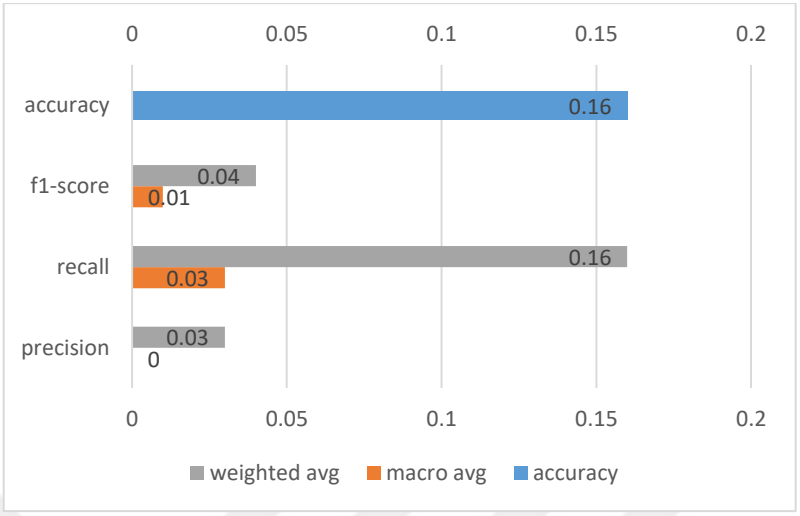

Figure 43: Results of ResNet 18 without the last four layers.

Table 9: ResNet 18 Parameter Information without the last four layers.

Total params	26,149,629
Trainable params	26,149,629
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	44.42
Params Size (MB)	99.75
Estimated Total Size (MB)	144.76

Now we will remove the last five layers of ResNet 18 and display accuracy as Figure 44 shows. It shows bad behaviour, with maximum accuracy reaching 21%.

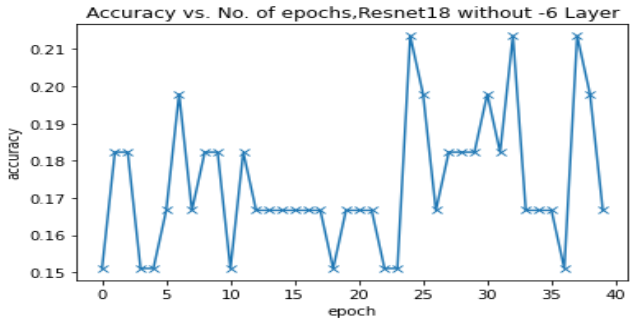

Figure 44: Accuracy of ResNet 18 without the last five layers with 40 epochs.

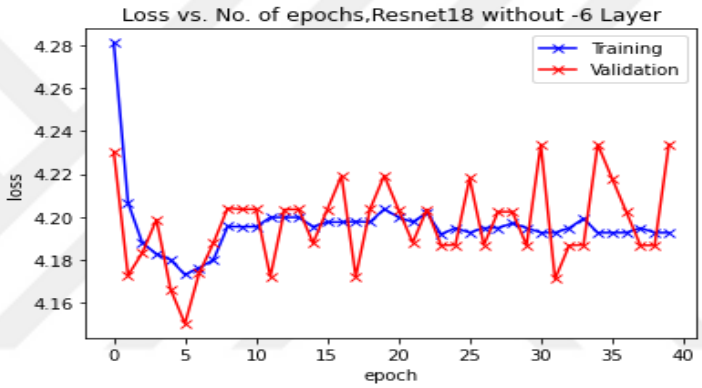

Figure 45: Loss of ResNet 18 without the last five layers with 40 epochs.

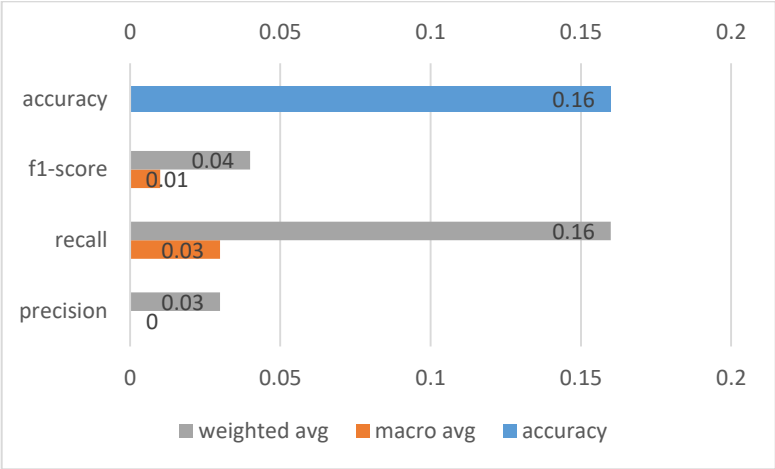

Figure 46: Results of ResNet 18 without the last five layers.

Table 10: ResNet 18 Parameter Information without the last five layers.

Total params	26,001,661
Trainable params	26,001,661
Non-Trainable params	0
Input Size (MB)	0.59
Forward/backward pass Size (MB)	22.21
Params Size (MB)	99.19
Estimated Total Size (MB)	121.99

Table 11 shows the accuracy, size, and average precision of ResNet 18 and other versions of it.

Table 11: ResNet 18 without last layers Accuracy, average precision, size and average response time in milliseconds.

Model	Size	Accuracy	Average Precision	Response Time in Milliseconds
ResNet18	110 MB	0.92	0.9	106 ms
ResNet18 Without AVG	126 MB	0.75	0.22	73 ms
ResNet18 with 15 Layer	102 MB	0.90	0.84	65 ms
ResNet18 with 10 Layer	111 MB	0.45	0.35	50 ms
ResNet18 with 5 Layer	144 MB	0.2	0.04	-
ResNet18 with 1 Layer	121 MB	0.21	0.05	-

Table 11 emphasized that ResNet 18 without removing any layers is better than other investigated versions. ResNet 18 without the last two layers comes in second place. As a result, we are going to use the original version of ResNet-18, i.e. we will not remove any of its layers.

In the next subsection, we will investigate the effect of changing the number of clusters, which can be specified by changing the cosine similarity threshold based on three evaluation matrices (Accuracy, Average Precision, and response time).

Now we will create clusters depending on similarity. If similarly = 0.69, then the number of clusters is 75. If similarly = 0.67, then the number of clusters is 66. If similarly = 0.66, then the number of clusters is 25. If similarly = 0.72, then the number of clusters is 170.

Table 12: Summary of ResNet 18 with different size clusters, Accuracy, Average Precision, and Average Response Time in a millisecond.

Number of Clusters	Accuracy	Average Precision	Response Time (millisecond)
Resnet18 With 170 Clusters	0.87	0.87	100
Resnet18 With 125 Clusters	0.92	0.9	106
Resnet18 With 75 Clusters	0.98	0.87	106
Resnet18 With 66 Clusters	0.97	0.91	102
Resnet18 With 25 Clusters	0.99	0.99	71

In the next subsection, we have replaced cosine similarity with L1 distance and investigated the effects of multiple values on the number of clusters and the overall performance. Here, when creating clusters depending on L1 Distance. If L1 = 360, then the number of clusters is 183. If L1= 355, then the number of clusters is 199. If L1 = 350, then the number of clusters is 266.

Table 13: Summary of ResNet 18 with different size clusters depending on L1 Distance, Accuracy, Average Precision, and Average Response time in milliseconds.

Model	Accuracy	Average Precision	Response Time (millisecond)
Resnet18 with 183 Clusters L1=360	0.85	0.8	98
Resnet18 with 199 Clusters L1=355	0.9	0.82	87
Resnet18 with 266 Clusters L1=350	0.75	0.72	102

As shown in Table 13, by using L1, the number of clusters is increasing which can affect the accuracy of Resnet-18. Cosine Similarity can be used to group a huge number of similar items together, and doesn't create a huge number of clusters, so it is suitable to use cosine similarity.

CHAPTER 4

4. LARGE-SCALE VISUAL PLACE RECOGNITION SYSTEM

A system called Vast Scale Visual Place Recognition has been developed as a way to search for a query image in a large collection of photographs when conventional approaches are impractical. The searching method depends on CNNs. There are two lines of CNN working together, one is for general queries, other is for specific queries. These lines cooperate together to institute the outcome of the system. CNN guarantees to search and compare the whole database without sequential iterating over all images. This yields a short-time response which is preferable for real-time systems. This system can apply to all large systems, while it is not dedicated to a specific type of image database. The system can be configurable to decrease or increase response time by determining the number of CNNs anticipating producing results, and it can boost or lower the precision in the same way.

4.1 Method.

The suggested method consists of two lines of ResNet-18 models. They are working together cooperatively to determine the right image for the inquired image. The first part determines the sequence of the image that might be the inquired image falls within. The second part resolves the precise place of that image. The system consists of three architectures:

1. The first part allocates the right sequence of images.
2. The second part allocates the precise place inside that sequence of images.
3. Investigation techniques allocate starting and stopping points of image sequences.

Figure 47 shows the system architecture. The first part consists of just one ResNet-18 architecture, while the second part consists of many ResNet-18 depending on the number of clusters found in the system.

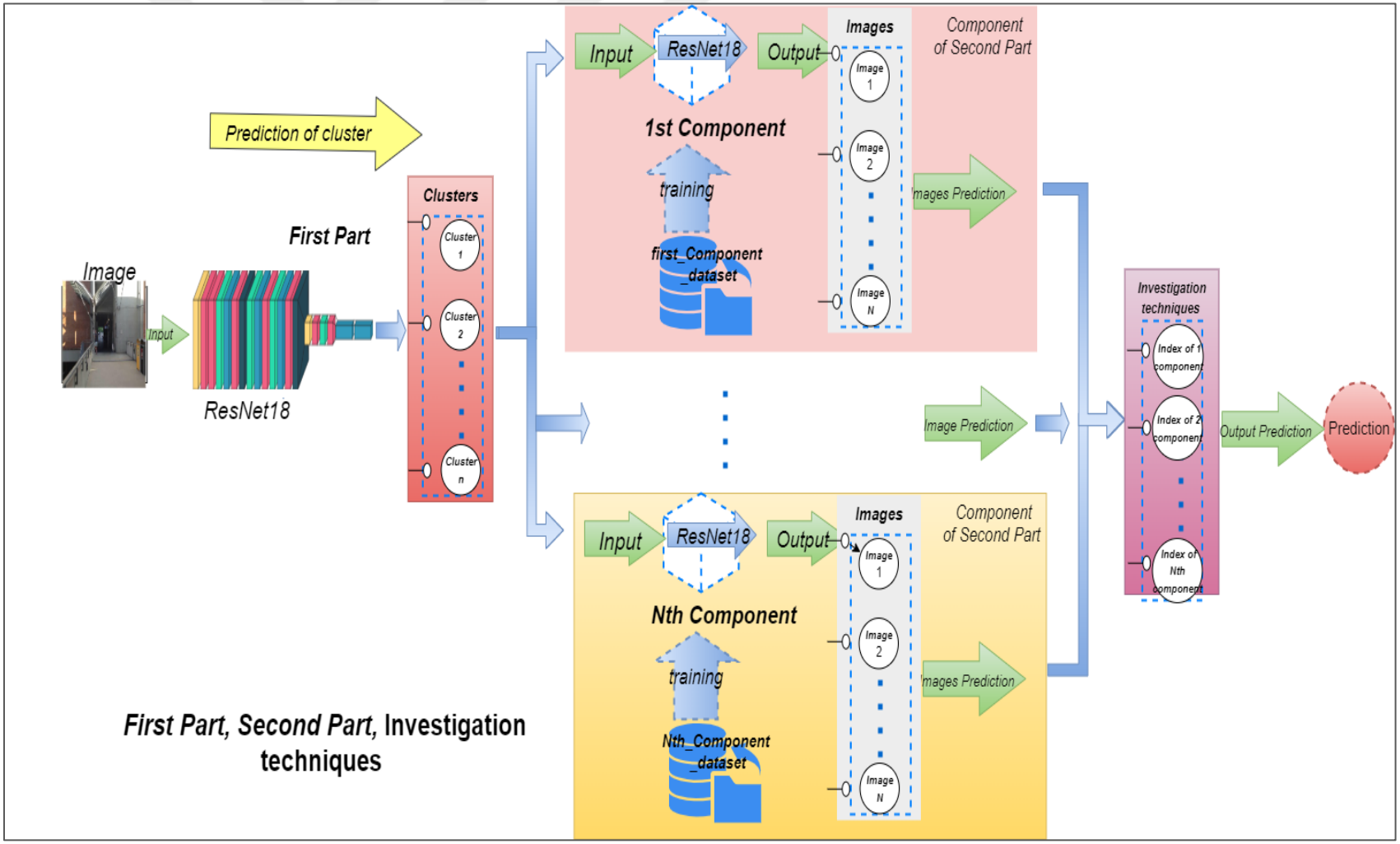

Figure 47: System Architecture.

4.2 Training the System

It is necessary to train the entire system. The first part is trained on all picture sequences. The first part was able to determine the correct order of photos for the requested image by the time the train had reached its destination. Each element in the second part is trained using the photos in its sequence. By the time the train is finished, each component should be aware of the proper picture cascade. Investigations methods reveal the second part's beginning and ending points to its constituent parts.

4.3 Expect Procedure

The Image with shape (3,227,227) is entered into the first part of the system, this first part determines the best component of the second part of the system to send this image to it because it highly probably knows the image much better. The correct component's assignment is an O(1) operation. If the system is set up to only activate one second part. Equation 9 is used to compute the total operation time. The complexity of searching in an array of n components is O(n), whereas the complexity of getting to the exact component in the array is O(1).

Operation time= ResNet18 response time of the first part + ResNet18 response time of the second line. (9)

Response time of ResNet-18 on Colab (Without GPU or TCP configuration) ≅100-120 milliseconds. (10)

If we substitute Equation (10) in Equation (9), the result will be Equation (11)

Operation time= 2 ×ResNet18 response time ≅2(100-120) ≅200-240 millisecond. (11)

If we increase the number of components of the second part to (n) so Equation (11) will be:

Operation time=n×ResNet18 response time≅ n(100-120) millisecond (12)

Now every component in the second part offers the best (3) choices. Following that, the cosine similarity operation is performed between the tested images and the whole choices to check the best matching, which will be the result of the system.

4.4 Multi-class Problem

Assigning each observation to one of the k classes is what the term "multi-class classification " alludes to (Wu et al., 2003). Numerous writers suggest using two-class classifiers for multi-class classification as two-class issues are significantly simpler to solve (Wu et al., 2003). Voting is a typical method for combining pairwise comparisons (Duan and Keerthi, 2005) (Wu et al., 2003). The class with the greatest number of successful two-class judgments is chosen after creating a rule for differentiating between every pair of classes (Wu et al., 2003). The voting process simply requires paired judgments, but it only forecasts a class label (Wu et al., 2003).

Recent visual recognition techniques' apparent success has primarily been attributed to how well they perform on classification problems since all potential classes are known at training time (Jain et al., 2014). But what about open set issues, where unidentified classes pop up during testing? Inferentially, even under the premise of partial class knowledge, we might reject the wide set of unknown classes if we could properly model just the positive data for any given class without overfitting (Jain et al., 2014).

The computer vision community is interested in the most recent classification results for the ImageNet Large-Scale Visual Recognition Challenge (Jain et al., 2014). With such low mistake rates (a convolutional neural network, which placed first in the 2013 ImageNet competition, achieved an error rate of 11.1%) (Jain et al., 2014). It is reasonable to question, however, if a situation in which all classes are known throughout the training period results in an appropriate evaluation of the general level of object recognition (Jain et al., 2014).

The detection results from the 2013 ImageNet challenge, which is significant, present a different picture (Jain et al., 2014). No method yields a result as striking as what

we find for classification: the best result is a mean average accuracy of merely 22.6% when unknown items must be discarded in the process of determining the location and label of a known object (Jain et al., 2014). When there is a chance of discovering fresh classes that weren't present in training during testing, detection belongs to the category of machine learning issues known as open set recognition (Jain et al., 2014).

The detection of novel classes, rejecting outlier or unknown classes, and/or at the same time detecting and recognizing known classes in the presence of unknown classes have all been examined in emerging research that goes beyond conventional binary models of True/False class association for open set recognition (Jain et al., 2014). These strategies are a fine place to start, but they do not directly address the main issue: multi-class open set recognition, where models should take into account multiple existing classes and offer the choice to either detect or reject fresh classes (Jain et al., 2014).

For applications requiring visual identification from an open collection, more potent binary classification models have been presented (Jain et al., 2014). The 1-vs-Set Machine (a dual-plane linear classifier) and W-SVM (a calibrated non-linear classifier) techniques are based on a formalization of the risk of the unknown in open set recognition provided by Scheirer et al (Jain et al., 2014).

By offering a per-class model utilizing only the positive data for each class, a one-class classifier like the one-class SVM looks to assist us to address the open set recognition problem (Jain et al., 2014). One-class classifiers do not make any assumptions regarding negative or unidentified classes, nor do they presume a closed environment (Jain et al., 2014). Unfortunately, one-class classifiers have difficulty maintaining separation between known positive and negative classes exactly because they do not utilize any negative data (Jain et al., 2014).

More theoretical research has been done on a multi-class problem, which becomes an obstacle in visual place recognition in evaluating VPR systems. To solve the problem and make it easy for future researchers, converting multi-class problems to two-class

problems is the main cornerstone to solving this dilemma. Two classes mean (Negative and Positive, True and False) problems, which is easy in performance metrics to be used. Metrics like: Accuracy, precision, recall, and Receiver operating characteristic (ROC). ROC graphs are helpful tools for choosing classification models based on how well they perform concerning the true positive rate (TPR) and false positive rate (FPR), which are calculated by adjusting the classifier's decision threshold (Raschka& Mirjalili, 2019).

A classification model is said to be poorer than random guessing if it falls below the diagonal of a ROC graph (Raschka& Mirjalili, 2019). A perfect classifier would have a TPR of 1 and an FPR of 0, placing it in the top-left corner of the graph (Raschka& Mirjalili, 2019). The performance of a classification model may then be described by computing the so-called ROC area under the curve (ROC AUC) based on the ROC curve (Raschka& Mirjalili, 2019).

Converting multi-classes problems to two-class problems can be done using some criteria. The first criterion is the radius, which is the range of expected class within, it places in the positive class, and otherwise, it would be in the negative class. How can this calculation be done?

The answer is first choosing the radius, some research called it tolerance range. Radius is the sanctioned range.

The second criterion is labelling the whole community the method works on. The labelling depends on the number of classes composing the dataset. Every element of the community should be assigned to one of the classes. Some elements presumably could have the same label, and some labels conceivably could have only one element.

The third criterion is the expecting method, which should effectuate precisely real assessments compatible with the labelling method and not exceed the labelling range, and not be beneath it.

After all of these criteria are fulfilled, converting a multi-class problem to a two-class problem can be done using the expectation method. If the expecting method anticipates the label of the inquired image within the range (true label Radius), it becomes a true expectation, so that the inquired element belongs to the positive class. If the expecting method anticipates the label of the inquired image out of the range (true label

Radius), it becomes a false expectation, so the inquired element belongs to the negative class.

4.5 Overfitting Problem

Overfitting is the term used when a model does not generalize properly from seen data to unobserved data (Ying, 2019). The model works well on the training set but fits badly on the testing set due to overfitting (Ying, 2019). This is because an over-fitted model struggles to deal with informational components in the testing set that may differ from those in the training set (Ying, 2019). As opposed to learning the discipline concealed in the data, over-fitted models tend to memorize all the data, including inevitable noise in the training set (Ying, 2019).

These phenomena may have several root causes. In general, we may divide them into three categories: 1) When the training set is too short, has less representative data, or has too many sounds, noise learning occurs (Ying, 2019). Due to this circumstance, what is heard can likely be learned and used to create predictions for the future (Ying, 2019). Therefore, a good algorithm should be able to discriminate between representative data and noise; 2) algorithm complexity: the trade-off in complexity, a crucial idea in statistical and machine learning, is a balance between Variance and Bias (Ying, 2019).

It speaks of striking a balance between precision and reliability (Ying, 2019). The model improves on average with less consistency when the algorithms are given too many hypotheses (or inputs) (Ying, 2019). Due to the fact that various datasets might have quite different models, multiple comparison procedures—which are common in induction algorithms and other Artificial Intelligence (AI) algorithms—may be used (Ying, 2019). The item with the highest score is always chosen throughout these procedures when comparing numerous things based on scores from an evaluation function (Ying, 2019). However, this procedure is likely to choose some things that will not increase classification accuracy or perhaps worsen it (Ying, 2019).

Numerous remedies based on various tactics are suggested to inhibit the various triggers to lessen the impact of overfitting (Ying, 2019). However, because of the

extensive usage of hypotheses, most of them perform badly when dealing with real-world problems (Ying, 2019). None of the hypothesis sets, however, can encompass every possible application area (Ying, 2019).

Early stopping is one of the solutions many researchers suggest (Ying, 2019). The "learning speed slow-down" problem is avoided by using this tactic (Ying, 2019). This problem suggests that noise learning causes algorithms' accuracy to cease increasing or even degrade after a certain point (Ying, 2019).

Network reduction as is well known, noise learning is a significant contributor to overfitting. Therefore, noise reduction makes sense as a study area for overfitting inhibition (Ying, 2019). Based on this reasoning, pruning is recommended to be used in relational learning, particularly in decision tree learning, to minimize the size of final classifiers (Ying, 2019). Pruning is a crucial idea that is used to minimize overfitting, increase classification accuracy, and reduce classification complexity by removing unnecessary or less useful data (Ying, 2019).

In many instances, especially in the field of supervised learning, the amount and quality of the training dataset can have a substantial impact on its success (Ying, 2019).

In training the first part and second parts of the framework, overfitting was the main problem in building the system, while the system showed good results in training and low results in testing. As many researchers suggested (Ying, 2019) increasing the training dataset is the solution for overfitting.

Increasing the training dataset can be done by many methods. One of these methods is to multiply the dataset images by adding noise to these images.

Flipping is one of the geometric operations that can be done to expand the image dataset (Shorten & Khoshgoftaar, 2019). More often than flipping the vertical axis, flipping the horizontal axis occurs (Shorten & Khoshgoftaar, 2019). One of the simplest

to employ, this expansion has been successful on datasets like CIFAR-10 and ImageNet (Shorten & Khoshgoftaar, 2019).

Colouring space is the typical way to encode digital picture data as a tensor of the dimensions (height, width, and colour channels) (Shorten & Khoshgoftaar, 2019). Another extremely doable technique is to do expansions in the space of the colour channels (Shorten & Khoshgoftaar, 2019).

By cropping the middle portion of each image, cropping may be utilized as a useful processing step for image data with mixed height and width dimensions (Shorten & Khoshgoftaar, 2019). Furthermore, random cropping may be employed to provide a result that is quite close to translations (Shorten & Khoshgoftaar, 2019).

Rotation Images are rotated right or left on an axis between 1° and 359° to perform rotation augmentations (Shorten & Khoshgoftaar, 2019). The rotation degree parameter has a significant impact on the safety of rotation augmentations (Shorten & Khoshgoftaar, 2019). On digit identification tests like MNIST, slight rotations like those between 1 and 20 or 1 to 20 might be helpful, but as the rotation degree grows, the label of the data is no longer preserved post-transformation (Shorten & Khoshgoftaar, 2019).

The translation is used to prevent positional bias in the data, shifting photos to the left, right, up, or down might be a very helpful adjustment (Shorten & Khoshgoftaar, 2019). For instance, if all the photos in a dataset are precisely centred, as is typical for face recognition datasets, the model would also need to be evaluated on those images (Shorten & Khoshgoftaar, 2019).

Injection of noise can be done by a matrix of random values, typically taken from a Gaussian distribution, and is used as part of noise injection (Shorten & Khoshgoftaar, 2019).

To deal with overfitting in the training data, these geometric transformations are a great option (Shorten & Khoshgoftaar, 2019).

Cutting at random is another intriguing Data expansion method. Random cutting, which is inspired by the mechanics of dropout regularization, is comparable to dropout except that it occurs in the input data space as opposed to being incorporated into the network design (Shorten & Khoshgoftaar, 2019).

Five ways of expanding training data are employed in this thesis. They are Gaussian noise, blur, histogram equalization, random cutting, random rotation, and random cutting. Five duplicate copies of each image are obtained by applying each method independently to the original. Now, the size of each training dataset used to train the system is increased by a factor of 5. The number of photos in the Nordland dataset is 55,184, multiplied by 5. The system will be trained on 275,920 photos in total.

4.6 Datasets for Evaluating Place Recognition

We will now assess the proposed system using benchmark datasets and contrast the results with those obtained using cutting-edge techniques.

4.6.1 Nordland dataset

A 728-kilometre tour in the spring, summer, fall, and winter yielded a brand-new dataset (Sünderhauf et al., 2013). Extreme seasonal variations are covered across about 3000 miles in both natural and artificial surroundings in around 40 hours of full-HD film (Sünderhauf et al., 2013). Furthermore, reliable real-world data is offered (Sünderhauf et al., 2013). They believe that this is now the biggest SLAM dataset available (Sünderhauf et al., 2013). The whole 10-hour voyage has been captured four times, once for each season, from the perspective of the train driver (Sünderhauf et al., 2013). As a result, the dataset may be thought of as consisting of single, 4-times-travelled, 728 km-long loops (Sünderhauf et al., 2013). The train travels through mostly rural areas, but it also travels through some cities and occasionally makes stops at train stations or signals (Sünderhauf et al., 2013). Figure 48 shows some images of the Nordland Dataset.

Figure 48: The four photos above depict the same location in the seasons of winter, spring, summer, and fall (Sünderhauf et al., 2013).

Khaliq et al. create a synthesized Nordland which consists of just 1221 for training and 1217 for referencing (Khaliq et al., 2019). Their method is called Region-VLAD (Khaliq et al., 2019). They compare their results with some methods like Cross-Region-BoW, Cross-Region-VLAD, FABMAP and others (Khaliq et al., 2019). They showed state-of-art performance over other methods as Figure 44 shows (Khaliq et al., 2019).

In our suggested method we train the system on the whole summer and winter Nordland dataset which consists of 55,184 images excluded tunnel images where the images are totally black. We tested the suggested method on 9,800 images of spring and fall images Nordland dataset. Where 4,900 images came from spring, and 4,900 images came from fall. Figure 50 shows our result with 84% AUC which is superior over other systems we compared within Figure 49. We triggered just one component of the second part. With an average total response time of 200-220 ms with $\pm$38 standard deviation, which is faster than most of the other systems as Table 14 will show.

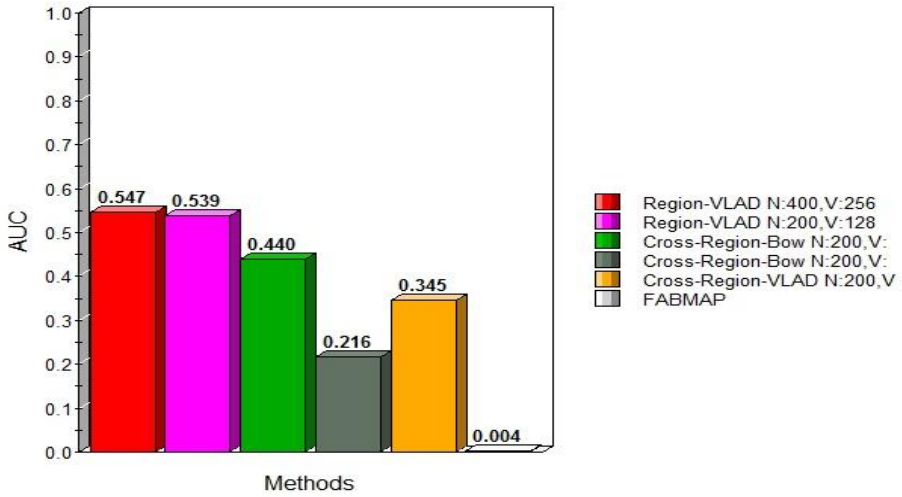

Figure 49: Results of different methods on the Nordland dataset (Khaliq et al., 2019).

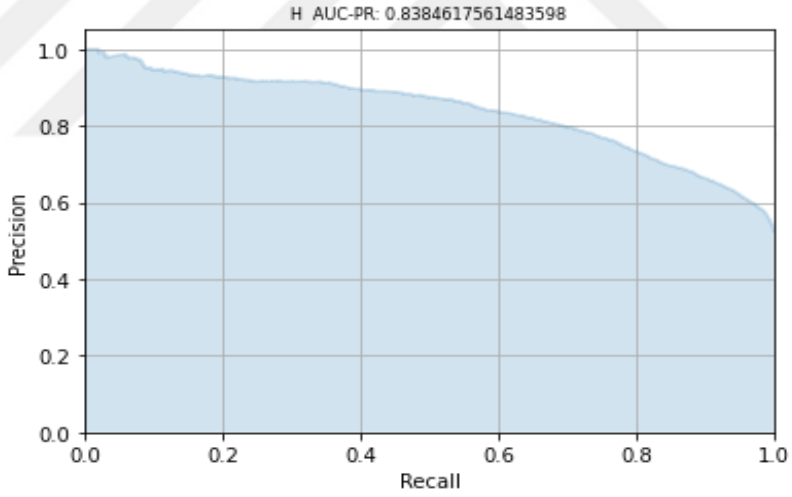

Figure 50: AUC Results of suggested method on Nordland Dataset.

4.6.2 Gardens Point dataset.

It was captured on the QUT campus in Brisbane and comprises two daytime treks and one nighttime journey, each lasting 200 photos (Hausler, Jacobson & et al., 2019). The right side of the walkways was used to capture the night trip, while the second-day trek and the one during the day were both recorded from the left side (Hausler, Jacobson & et al.,

2019). On both Gardens Point treks, there is more intense lightning (Khaliq et al., 2019). Gardens Point is a challenging dataset due to strong viewpoint and condition variation (Khaliq et al., 2019). Figure 51 shows some samples of the Gardens Point Dataset.

Figure 51: Samples of the Garden Point dataset's image pairings.

The proposed system was trained on day-left photos of the Gardens Point dataset. The Nighttime pictures are employed in testing. The Gardens Point dataset consists of sequences of images, so it is suitable for VPR systems. The challenge of this dataset is the strong illumination variation between day and night, and the strong variation in the viewpoint of the image near 180 °. The experiment has been conducted by grouping images by the similarity of 60%, then the system was trained as usual sequence datasets. We triggered 7 components of the second part in one experiment, and 1 component of the second part in another experiment. The average response time for the system with one component triggered is between 200-220 milliseconds on average. AUC was 0.85 for 7 components triggered which is better than Region-VLAD (Khaliq et al., 2019), Cross-region-BoW, Cross-region-VLAD, AlexNet365 and HybridNet SPP. AUC for 1 component triggered was 0.64 which is near the state-of-the-art Region-VLAD-N:200-V:128 method by Khaliq et al., which was 0.65. These results show cutting-edge performance in the Visual place recognition research area. Figure 52 shows the AUC PR-Curve.

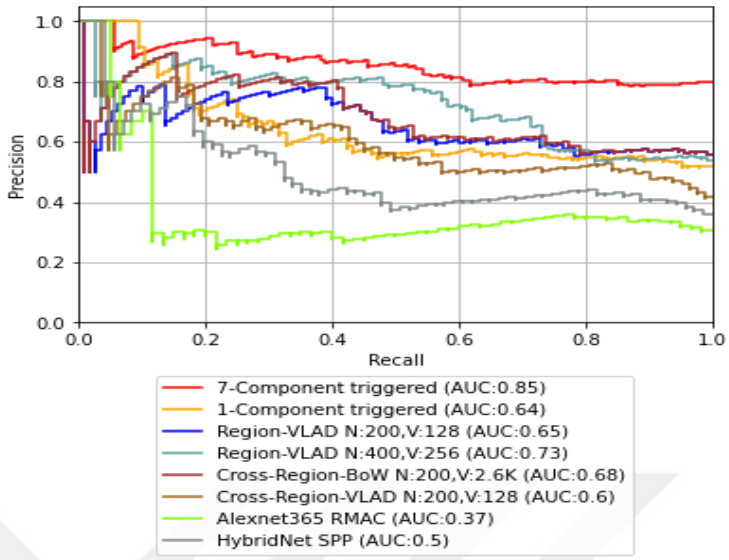

Figure 52: AUC PR-curves for Gardens Point Dataset.

4.6.3 Berlin-A100 dataset.

It may be obtained from Mapillary, a site for crowdsourcing (Lu et al., 2021). It comprises two traverses of the same road with some differences brought on by dynamic vehicles as well as moderate appearance and viewpoint alterations (Lu et al., 2021). Additionally, there is a shift in speed between the two traverses (Lu et al., 2021). The challenge of this dataset is the pedestrians and cars excluded some part of the background of the images, which mainly consists of places to be recognized. The experiment has been conducted by grouping images by the similarity of 70%, then the system was trained as usual sequence datasets. Figure 53 shows some samples of the Berlin-A100 Dataset.

Figure 53: Samples of the Berlin-A100 dataset's image pairings.

We triggered 7 components of the second part in one experiment, and 1 component of the second part in another experiment. The average response time for the system with one component triggered is between 200-220 milliseconds on average. AUC was 0.78 for 7 components triggered which is better than Region-VLAD (Khaliq et al., 2019), Cross-Region-Bow N:200-V 2.6 K, Cross-region-VLAD, AlexNet365 and HybridNet SPP. The AUC of 7 components triggered is the same as the result of the best on this dataset Cross-Region-Bow N:200-V 10 K, which was 0.87. AUC for 1 component triggered was 0.44. These results show cutting-edge performance in the challenge dataset. Figure 54 shows the AUC PR-Curve.

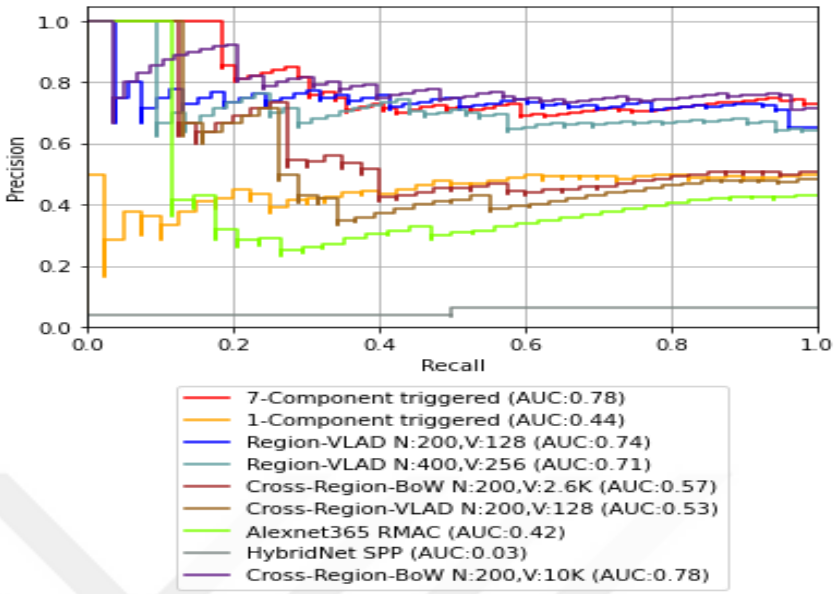

Figure 54: AUC PR-curves for Berlin-A100 Dataset.

4.6.4 Berlin Halenseestrasse dataset.

This dataset, which was also collected from Mapillary, includes daylight trips made in an urban region by a vehicle user (reference) and a bicycle user (query) (Camara and Přeučil, 2020). Given that the query photographs were shot from the bicycle's lane, viewpoint alterations are significantly more pronounced than for the prior collection (Camara and Přeučil, 2020). Such lanes occasionally have foliage surrounding them, which makes it more difficult to locate many similarities with reference pictures (Camara and Přeučil, 2020). As many as 10 question photos might be rotated horizontally, either by 90 degrees clockwise or anticlockwise or entirely upside down, which has a severe impact on recognition (Camara and Přeučil, 2020).

The viewpoint switches between a camera installed behind the windscreen of a moving automobile in the Berlin Halenseestrasse dataset and a camera mounted on a bicycle in the cycle lane next to the road (Neubert and Protzel, 2016). Additionally, the sun's varying positions cause the light to shift (Neubert and Protzel, 2016). Figure 55 shows some samples of the Berlin-Halenseestrasse Dataset.

Figure 55: Samples of the Berlin-Halenseestrasse dataset's image pairings.

The experiment has been conducted by grouping images by the similarity of 65%, then the system was trained as usual sequence datasets. We triggered 7 components of the second part in one experiment, and 1 component of the second part in another experiment. The average response time for the system with one component triggered is between 200-220 milliseconds on average. AUC was 0.64 for 7 components triggered. AUC for 1 component triggered was 0.5. The best outcome is produced by Region-VLAD by Khaliq et al., followed by the 7-triggered component in second place with 0.64 AUC. Due to the homogeneity of the majority of the locations in the Berlin Halenseestrasse dataset, our suggested approach produces reasonably decent results. Figure 56 shows the AUC PR-Curve.

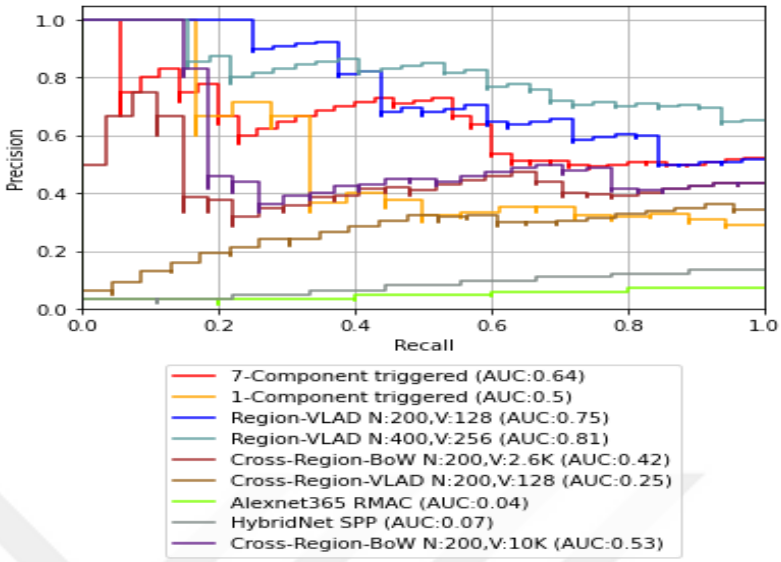

Figure 56: AUC PR-curves for Berlin-Halenseestrasse Dataset.

4.6.5 Berlin Kudamm dataset.

This dataset, whose source is once more Mapillary, is typically regarded as the hardest of the five under consideration (Camara and Přeučil, 2020). Reference frames were shot from the top deck of a bus, while question frames were taken from a bicycle, creating extremely significant perspective alterations and modest condition differences in this bustling metropolitan setting (Camara and Přeučil, 2020). Numerous dynamic items, such as moving and parked automobiles, people, and bicycles, are present in addition to these changes (Camara and Přeučil, 2020). Additionally, there is a lot of foliage that is typically present and obscures more static and distinctive objects like buildings (Camara and Přeučil, 2020).

Perceptual aliasing occurs in this dataset when homogenous situations, such as moving objects and dynamic distractions like cars and people, are confused with them (Keetha et al., 2021). Complexity is increased by the significant viewpoint change in particular (Keetha et al., 2021). Figure 57 shows some samples of the Berlin-Kudamm Dataset.

Figure 57: Samples of the Berlin-Kudamm dataset's image pairings.

The experiment has been conducted by grouping images by the similarity of 67%, then the system was trained as usual sequence datasets. We triggered 7 components of the second part in one experiment, and 1 component of the second part in another experiment. The average response time for the system with one component triggered is between 200-220 milliseconds on average. AUC was 0.33 for 7 components triggered. AUC for 1 component triggered was 0.33. The best results are displayed by Cross-Region-Bow, followed by Region-VLAD. The third place goes to the proposed system, with 1-component triggered and 7-component triggered. The Berlin Kudamm dataset includes objects that move and are confusing, such as automobiles, people, and trees (Zaffar et al. 2019). Most features of the image are not significant to place-centric CNN since they do not take into account moving objects. All VPR methods have a substantial barrier in Berlin Kudamm because of the ambiguous items and homogeneous landscapes that result in perceived aliasing (Zaffar et al. 2019). Our suggested approach produces reasonably decent results. Figure 58 shows the AUC PR-Curve.

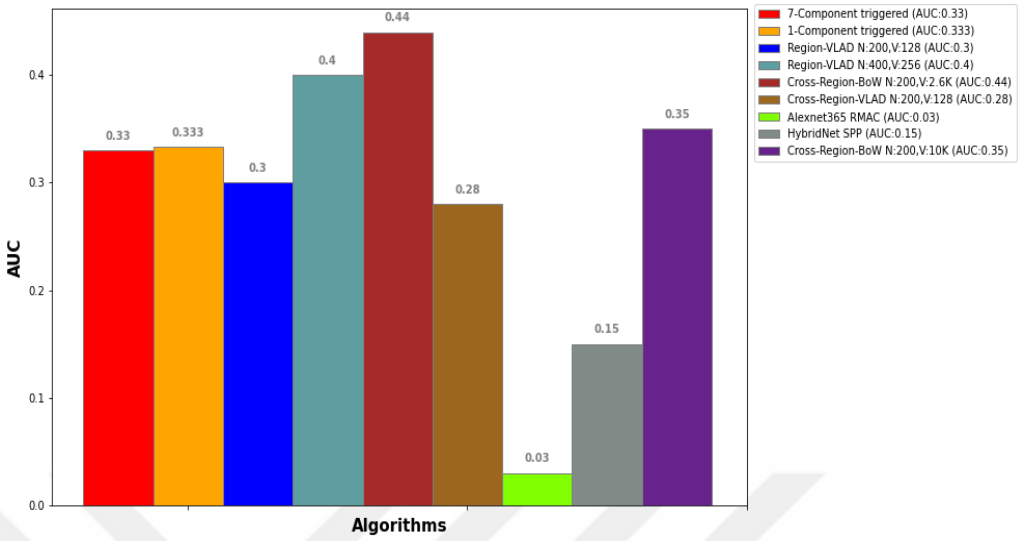

Figure 58: AUC PR-curves for Berlin-Kudamm Dataset.

4.6.6 AmsterTime dataset.

AmsterTime is a difficult dataset for visual place recognition benchmarking in the midst of a significant domain change (Yildiz et al., 2022). AmsterTime provides a collection of 2,500 expertly picked photographs that are linked to the same scenario from an Amsterdam street view and historical archive image data (Yildiz et al., 2022). The image pairings show the same location from several angles, perspectives, and appearances (Yildiz et al., 2022). AmsterTime is a GIS navigation tool that is directly crowdsourced, as opposed to other benchmark datasets (Mapillary) (Yildiz et al., 2022). 1200+ pictures available without a license from the Amsterdam City Archive, depicting urban areas in the city of Amsterdam during the past 100 years (Yildiz et al.,2022). Images captured with different cameras, time delays, structural changes, occlusion, viewpoints, appearances, and illuminations in the archive and street view image pairs show the same location (Yildiz et al., 2022). When evaluating the learnt similarity features with class activation mapping frameworks like Grad-CAM, relevant landmarks from Amsterdam city are compiled into a new classification dataset from the Google Landmarks dataset (Yildiz et al., 2022). As shown by the baseline findings, AmsterTime includes the entire temporal coverage of Amsterdam city with a significant domain change between query and gallery, making it

particularly difficult to assess VPR models (Yildiz et al., 2022). Figure 59 shows samples of the AmsterTime dataset's image pairings.

Figure 59: Samples of the AmsterTime dataset's image pairings. Extreme occlusions, shifting viewpoints, camera lens aberrations, and color shifts present difficulties (Yildiz et al., 2022).

The proposed system was trained on 1231 new images of AmsterTime. The test was performed on archival images. The challenge of this dataset is no sequences of images are presented, just a collection of images. But we experimented by grouping images by similarity, then the system was trained as usual sequence datasets. We triggered 7 components of the second part. The average response time for the system was 1021 milliseconds. AUC was 0.29 as figure 60 shows. The outcome is a good result, while AmsterTime is a challenging dataset with no sequence of images. There is no spatially linked between images, so any method will be difficult to generate accurate results. Figure 60 shows AUC for AmsterTime.

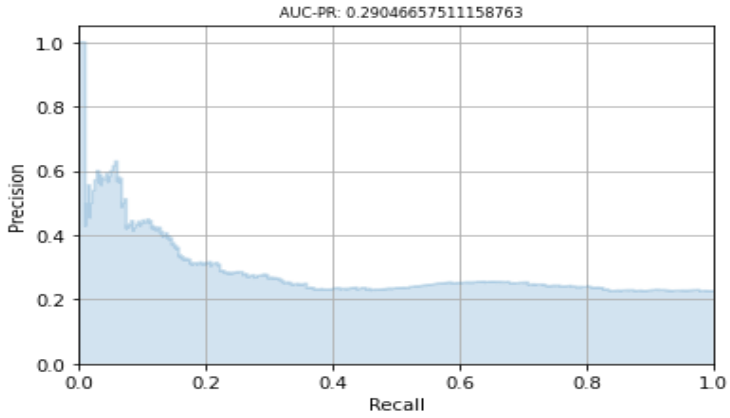

Figure 60: AUC PR-curves for AmsterTime Dataset.

4.7 Comparing Time.

Comparing time is an important metric in evaluating any system. Our proposed system shows good results in benchmark datasets, but now the speed of the system according to other state-of-art methods should be checked. Comparing the proposed system time with the time of state-of-art methods, showing our system achieved 200-220 milliseconds compared to the best of showed methods, Region VLAD with 446 milliseconds, shows the superior performance of our proposed system over other methods in time. Table 14 shows state-of-art results. Time is calculated in milliseconds.

Table 14: Total time for different methods.

Method	Platform	Total time (ms)	
Cross-Region-BoW	Titan X Pascal GPU	415	(Zhang et al., 2021)
Region-VLAD	(GPU/CPU) NVIDIA P100 / Intel Xeon Gold 6134 @3.2 GHz	446.405 - 533.245	(Khaliq et al., 2019)
On the Performance of ConvNet Features for Place Recognition	-	204	(Zhang et al., 2021)
LeNet	(standard CPU)	400	(Zhang et al., 2021)
A Hybrid Compact Neural Architecture for Visual Place Recognition	-	60	(Zhang et al., 2021)

4.8 Discussion

The suggested system exhibits cutting-edge performance, because of its design, and is made up of several ResNet18 models overall. Together, these elements are producing the system's optimum outcome in the least amount of time possible. The system was trained in diverse appearances, seasonal changes, and viewpoint shifts using training approaches, which led to high performance. The pre-trained ResNet 18 on Places-365 is adequate for visual place identification, according to the system. The suggested might be used in multi-agent robot systems, where a primary robot could be the first part to receive a requested image and additional robots could serve as the second part of numerous components. Multi-agent systems are depicted in Figure 61.

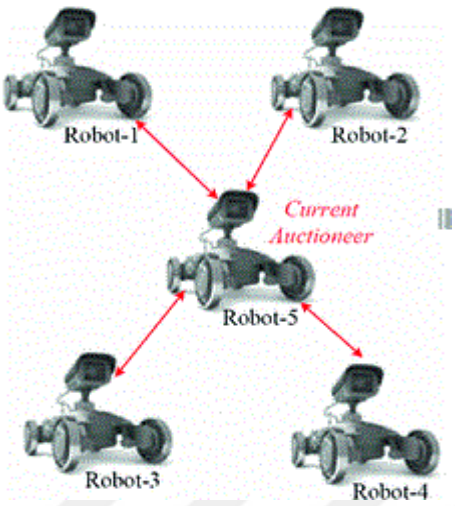

Figure 61: Multi-agent Robot.

This approach may also be utilized in search engine systems where a central computer can choose as the first part and sub-computers as the second part. Where a central computer receives an image inquiry, then it can decide which components of the second part are responsible for that image. That component of the second part could accurately judge the location of a queried image as shown in Figure 62.

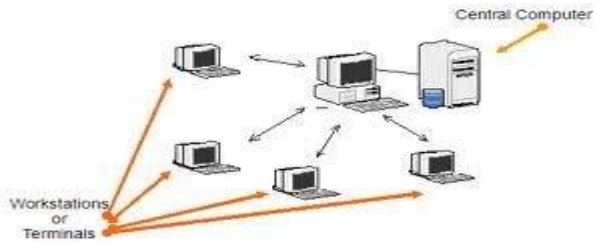

Figure 62: Images search engine system.

CHAPTER 5

5. Conclusion

The thesis presented the efficacy of utilizing clustering techniques for organizing photo collections to facilitate efficient and effective searches in sizeable visual place recognition databases. Searching for relevant photos in large-scale databases is a challenging task owing to the prolonged search operations and the need for rapid comparison computations to extract the best results. Therefore, search techniques serve as the primary differentiator among various studies and their strategies for searching through massive picture databases.

In this thesis, the system was built using two components. The photos of the datasets that were grouped using the various approaches described in the thesis constructed the first part of the proposed approach. The second part was the employment of several CNN architectures that worked together to conduct searches inside clusters of pictures.

A variety of picture grouping techniques were contrasted. We agreed that AlexNet's findings were superior to HOG's findings, while SIFT produces poor outcomes. However, HOG was more efficient than AlexNet since it produced results more rapidly, while AlexNet took longer to reply and generate results.

The primary block of CNN's multi-models was carefully selected because it forms the foundation of the whole system. Numerous comparison tasks had been carried out between RESNET-18 and AlexNet using time and accuracy criteria. RESNET-18 demonstrated the best performance, despite AlexNet's quicker reaction time. Here, accuracy was taken into account first as the search system had to produce accurate results. In addition, the time came in second consideration, as it is still a crucial need for the alleged system that needed to be met.

The system's framework was mostly dependent on RESNET-18, as it produced the greatest results. Two lines of RESNET-18 made up the majority of the recommended system. In the first part, one RESNET-18 architecture is used, and several components of the RESNET-18 architecture were included in the second part. The best cluster to which the questioned image belongs had to be assigned by the first part. The second part's tasks included precisely identifying the ID of the image under examination. The Indexing system, which made it easier to create the rectified index ID of the second part results, made up the third component of the framework.

In this thesis, we addressed the issue that most Visual Place Recognition systems encounter, which is a direct relationship between search community size and search time. The search procedures take a long time as the dataset size increases. After the thesis has been authorized, we may create a system that uses a sizable dataset and continuous searching. Search engines and real-time systems could perform better if search activities are stable.

The thesis has produced state-of-the-art results in the total time to generate the result of the requested image, between 200-220 ms. it may represent a huge advancement in the field of visual place recognition. In addition, The system that was suggested in this thesis outperformed numerous cutting-edge techniques on a variety of difficult benchmark datasets.

REFERENCES

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5297-5307).

Baumgartl, H., & Buettner, R. (2020). Development of a highly precise place recognition module for effective human-robot interactions in changing lighting and viewpoint conditions.

Bay, H., Tuytelaars, T., & Gool, L. V. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010, September). Brief: Binary robust independent elementary features. In *European conference on computer vision* (pp. 778-792). Springer, Berlin, Heidelberg.

Camara, L. G., & Přeučil, L. (2020). Visual place recognition by spatial matching of high-level CNN features. *Robotics and Autonomous Systems*, *133*, 103625.

Chen, Z., Lam, O., Jacobson, A., & Milford, M. (2014). Convolutional neural network-based place recognition. *arXiv preprint arXiv:1411.1509*.

Chen, Z., Jacobson, A., Sünderhauf, N., Upcroft, B., Liu, L., Shen, C., ... & Milford, M. (2017, May). Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3223-3230). IEEE.

Chen, Z., Maffra, F., Sa, I., & Chli, M. (2017, September). Only look once, mining distinctive landmarks from convnet for visual place recognition. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 9-16). IEEE.

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, **27(6)**, 647-665.

Duan, K. B., & Keerthi, S. S. (2005, June). Which is the best multiclass SVM method? An empirical study. In *International workshop on multiple classifier systems* (pp. 278-285). Springer, Berlin, Heidelberg.

Facil, J. M., Olid, D., Montesano, L., & Civera, J. (2019). Condition-invariant multi-view place recognition. *arXiv preprint arXiv:1902.09516*.

Fazl-Ersi, E., & Tsotsos, J. K. (2012). Histogram of oriented uniform patterns for robust place recognition and categorization. *The International Journal of Robotics Research*, ***31*(4)**, 468-483.

Gálvez-López, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, ***28*(5)**, 1188-1197.

Garg, S., Harwood, B., Anand, G., & Milford, M. (2020). Delta descriptors: Change-based place representation for robust visual localization. *IEEE Robotics and Automation Letters*, ***5*(4)**, 5120-5127.

Garg, S., Suenderhauf, N., & Milford, M. (2022). Semantic–geometric visual place recognition: a new perspective for reconciling opposing views. *The International Journal of Robotics Research*, ***41*(6)**, 573-598.

Han, J., Kamber, M., & Pei, J. (2012). Data mining concepts and techniques third edition. University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University.

Hausler, S., Jacobson, A., & Milford, M. (2019). Multi-process fusion: Visual place recognition using multiple image processing methods. *IEEE Robotics and Automation Letters*, ***4*(2)**, 1924-1931.

Hausler, S., Jacobson, A., & Milford, M. (2019, November). Filter early, match late: Improving network-based visual place recognition. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3268-3275). IEEE.

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

Jain, L. P., Scheirer, W. J., & Boult, T. E. (2014, September). Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision* (pp. 393-409). Springer, Cham.

Jégou, H., Douze, M., Schmid, C., & Pérez, P. (2010, June). Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 3304-3311). IEEE.

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., & Schmid, C. (2011). Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, ***34*(9)**, 1704-1716.

Keetha, N. V., Milford, M., & Garg, S. (2021). A hierarchical dual model of environment- and place-specific utility for visual place recognition. *IEEE Robotics and Automation Letters*, *6*(4), 6969-6976.

Khaliq, A., Ehsan, S., Chen, Z., Milford, M., & McDonald-Maier, K. (2019). A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes. *IEEE transactions on robotics*, *36*(2), 561-569.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, ***60*(6)**, 84-90

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

Lowry, S., & Milford, M. J. (2016). Supervised and unsupervised linear learning techniques for visual place recognition in changing environments. *IEEE Transactions on Robotics*, ***32*(3)**, 600-613.

Lu, F., Chen, B., Zhou, X. D., & Song, D. (2021). STA-VPR: Spatio-temporal alignment for visual place recognition. *IEEE Robotics and Automation Letters*, *6*(3), 4297-4304.

Neubert, P., & Protzel, P. (2016). Beyond holistic descriptors, keypoints, and fixed patches: Multiscale superpixel grids for place recognition in changing environments. *IEEE Robotics and Automation Letters*, *1*(1), 484-491.

Nister, D., & Stewenius, H. (2006, June). Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (**Vol. 2**, pp. 2161-2168). Ieee.

Olid, D., Fácil, J. M., & Civera, J. (2018). Single-view place recognition under seasonal changes. *arXiv preprint arXiv:1808.06516*.

Oliva, A. (2005). Gist of the scene. In *Neurobiology of attention* (pp. 251-256). Academic press.

Prykhodchenko, R., & Skruch, P. (2022). Road scene classification based on street-level images and spatial data. *Array*, 100195.

Raschka, S., & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd.

Razavian, A. S., Sullivan, J., Carlsson, S., & Maki, A. (2016). Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, ***4*****(3)**, 251-258.

Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 806-813).

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1-48.

Sünderhauf, N., Neubert, P., & Protzel, P. (2013, May). Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons. In *Proc. of workshop on long-term autonomy, IEEE international conference on robotics and automation (ICRA)* (p. 2013).

Sünderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., & Milford, M. (2015). Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Robotics: Science and Systems XI*, 1-10

Warren, M., McKinnon, D., He, H., & Upcroft, B. (2010). Unaided stereo vision based pose estimation. In G. Wyeth & B. Upcroft (Eds.), Australasian conference on robotics and automation. Australian Robotics; Automation Association. http://eprints.qut.edu.au/39881/

Wu, T. F., Lin, C. J., & Weng, R. (2003). Probability estimates for multi-class classification by pairwise coupling. Advances in Neural Information Processing Systems, 16.

Ying, X. (2019, February). An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, No. 2, p. 022022). IOP Publishing.

Yildiz, B., Khademi, S., Siebes, R. M., & van Gemert, J. (2022). AmsterTime: A Visual Place Recognition Benchmark Dataset for Severe Domain Shift. *arXiv preprint arXiv:2203.16291*.

Zhang, X., Wang, L., & Su, Y. (2021). Visual place recognition: A survey from deep learning perspective. Pattern Recognition, 113, 107760.

Implementing AlexNet CNN Architecture Using TensorFlow 2.0+ and Keras.Available at: https://towardsdatascience.com/implementing-AlexNet-cnn-architecture-using-tensorflow-2-0-and-keras-2113e090ad98 Accessed 19.11.2022.

AlexNet.Mathworks.Available at: https://www.mathworks.com/help/deeplearning/ref/AlexNet.html Accessed 19.11.2022.

Image Category Classification Using Deep Learning.Mathworks.Available at: https://www.mathworks.com/help/vision/ug/image-category-classification-using-deep-learning.html#responsive_offcanvas Accessed 19.11.2022.

Taxicab geometry.Wikipedia.Available at: https://en.wikipedia.org/wiki/Taxicab_geometry Accessed 19.11.2022.

SIFT - Scale-Invariant Feature Transform.Prof. Dr. Edmund Weitz.Available at: http://weitz.de/sift/ Accessed 19.11.2022.

SIFT(Scale-invariant feature transform).Towards Data science. Available at: https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37 Accessed 19.11.2022.

Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor.Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/ Accessed 19.11.2022.