ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Serkan ÖZTÜRK

THE EFFECT OF FEATURE SELECTION METHODS ON DEEP CONVOLUTIONAL NEURAL NETWORKS

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2018

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

THE EFFECT OF FEATURE SELECTION METHOD ON DEEP CONVOLUTIONAL NEURAL NETWORKS

Serkan ÖZTÜRK

MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was received and approved the award of degree of the Master of Science by the board of jury on 08/08/2018.

....	....	....
Asst. Prof. Dr. B. Melis ÖZYILDIRIM	Assoc. Prof Dr. Zekeriya TÜFEKÇİ	Asst. Prof. Dr. Fatih ABUT
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institude of Natural and Applied Science

ABSTRACT

MSc THESIS

THE EFFECT OF FEATURE SELECTION METHODS ON DEEP CONVOLUTIONAL NEURAL NETWORKS

Serkan ÖZTÜRK

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
Year: 2018, Pages: 112
Jury : Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
: Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
: Asst. Prof. Dr. Fatih ABUT

Since there has been large increase in the amount of data in recent years, and these data have become increasingly higher dimension, it has become even more important to extract discriminative high-quality features for this data. Specifically, in image classification, Deep Convolutional Neural Networks are frequently used because of their high discrimination power. However, the dimension of the features learned by Deep Convolutional Neural Networks are very high and may include irrelevant or unnecessary features for classification tasks. Removing these irrelevant and unnecessary features may lead to increased classification performance. Although there are a lot of publications about the Convolutional Neural Networks in the literature, there are not many publications on the selection of the features extracted by them.

In this thesis, various feature selection methods have been applied to the features obtained from two fully connected layers of well-known deep neural network architectures such as AlexNet, VGG16 and VGG19, and the classification performances of these selected features have been compared.

Key Words: Deep learning, deep convolutional neural networks, feature selection, AlexNet, VGG Net

ÖZ

YÜKSEK LİSANS TEZİ

DERİN EVRİŞİMSEL SİNİR AĞLARINDA ÖZELLİK SEÇİM YÖNTEMLERİNİN ETKİSİ

Serkan ÖZTÜRK

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Dr. Öğr. Üyesi Buse Melis ÖZYILDIRIM
Yıl: 2018, Sayfa 112
Jüri : Dr. Öğr. Üyesi Buse Melis ÖZYILDIRIM
: Doç. Dr. Zekeriya TÜFEKÇİ
: Dr. Öğr. Üyesi Fatih ABUT

Son yıllarda veri miktarında büyük artış olması ve bu verilerin gittikçe daha yüksek boyutlu hale gelmesi ile birlikte, veriler için ayrım gücü yüksek özelliklerin çıkarımı daha da önemli hale gelmiştir. Özellikle görüntü sınıflandırmada, bu ayrım gücü yüksek özellikleri öğrenmesi nedeniyle Derin Evrişimsel Sinir Ağları sıklıkla kullanılmaktadır. Ancak Derin Evrişimsel Sinir Ağlarının öğrendiği özelliklerin boyutu çok yüksektir ve sınıflandırma görevi için ilgisiz veya gereksiz özellikleri de içerebilir. Bu ilgisiz ve gereksiz özelliklerin kaldırılması sınıflandırma performansının artmasına neden olabilir. Literatürde Evrişimsel Sinir Ağları hakkında çok fazla yayın olmasına rağmen, onlar tarafından çıkarılan özelliklerin seçilmesi hakkında çok fazla yayın bulunmamaktadır.

Bu tezde, AlexNet, VGG16 ve VGG19 gibi iyi bilinen derin sinir ağı mimarilerinin tam bağlı iki katmanından elde edilen özelliklere çeşitli özellik seçimi yöntemleri uygulanmış ve seçilen bu özelliklerin sınıflandırma performansları karşılaştırılmıştır.

Anahtar Kelimeler: Derin öğrenme, derin evrişimsel sinir ağları, özellik seçimi, AlexNet, VGG Net

GENİŞLETİLMİŞ ÖZET

Günümüzde, İnternet'in dünya genelinde yayılımıyla birlikte, çok miktarda veri dijital ortamda depolanmaktadır. Depolanan bu veriler, elde edildiği cihazın mekanik problemleri, depolandığı yerin gayrı resmi olması durumunda orjinali üzerinde oynanması gibi çeşitli nedenler ile çok sayıda gürültü içermektedir. Bu verileri kullanarak yapılan makine öğrenimi yöntemlerinin bu gürültü ile baş edebilmesi önemlidir. Bu amaçla, makine öğrenimi yöntemlerinin çoğunu kullanmadan önce bir ön işlem aşaması olarak özellik seçim yöntemleri uygulanmaktadır.

Makine öğrenimi yöntemlerinin baş etmesi gereken diğer bir zorluk da verilerin yüksek boyuta sahip olmasıdır. Yüksek boyut, makine öğrenimi yöntemlerinin eğitimi esnasında ezberleme olarak adlandırılan ve makine öğrenme yöntemlerinin test verilerine karşı düşük doğruluk değerleri ile sonuç üretmesine neden olan bir sürece neden olabilmektedir. Boyut laneti olarak adlandırılan bu durumu önlemek için ya çok daha fazla eğitim verisi kullanılmalı ya da boyut azaltılmalıdır. Boyut azaltmak için genel olarak iki yaklaşım kullanılmaktadır. Birincisi özellik çıkarımı, diğer ise özellik seçimidir.

Özellik çıkarma yöntemleri mevcut özelliklerin tamamını kullanarak çeşitli dönüşümlerin birbiri ardına gerçeklenmesi ile yeni bir özellik kümesi elde eder. Elde edilen bu özellik kümesinin boyutu orjinal olandan daha küçüktür ve elemanları orjinal özellik kümesinin elemanlarından farklıdır. Özellik seçimi yöntemleri, makine öğrenimi yöntemlerinin doğruluk değerinin yükselmesine ya da aynı kalmasına neden olan, orjinal özellik kümesinden ayrım gücü en yüksek olan özellikleri seçme işlemidir. Özellik seçim yöntemleri genel olarak dört adım içermektedir: alt küme üretme prosedürü, alt küme değerlendirme prosedürü, durma kriteri ve doğrulama prosedürü.

Alt küme üretme prosedürü, temel olarak bir arama prosedürüdür. Özellik kümesinin boyutuna göre tam arama, sezgisel arama ya da rastgele arama

yöntemlerini kullanarak alt küme üretme işlemlerini yerine getirir. Elde edilen bu alt kümelerden optimal olanı bulmak için bir değerlendirme prosedürü yürütülür. Değerlendirme fonksiyonları, uzaklık, bilgi, bağlılık, tutarlılık veya sınıflandırıcı hata oranı ölçümleri gibi çeşitli kriterler ile alt kümenin ayrım gücünü ölçer. Özellik seçimi sürecinin makul bir sürede gerçeklenmesi için arama ve değerlendirme prosedürlerinin belirli bir durma kriterine sahip olması önemlidir. Daha sonra elde edilen özellik alt kümesi bir test kümesi ile değerlendirilir.

Özellik seçim yöntemleri sınıflandırıcı yöntem ile arasındaki ilişkiye göre üç kategoriye ayrılabilir: filtre, sarmalayıcı ve gömülü. Filtre yöntemleri sınıflandırıcı algoritmasından bağımsız olarak, uzaklık, bağımlılık, tutarlılık ve bilgi ölçümleri ile özellik seçmeyi içerir. Sarmalayıcı yöntemler ise üretilen her özellik alt kümesinin bir sınıflandırıcı algoritma ile değerlendirmesini içerir. Sarmalayıcı yöntemler filtre yöntemlerine göre doğruluk değerleri yüksek yöntemler olmasına karşın hesaplama maliyeti daha yüksektir. Gömülü yöntemler ise belirli bir sınıflayıcı algoritmanın parçası olarak işlem görürler ve bu algoritmaya özeldirler.

Özellikle görüntü verileri ile çalışırken, makine öğrenmesinde ele alınması gereken bir başka zorluk veriyi temsil eden bir özellik kümesi elde etmektir. Çoğu makine öğrenimi algoritması için özellik kümesi uzman görüşüne dayanarak elle elde edilir. Bu durum hem zahmetlidir hem de hataya açıktır. Özellik kümesini kendi başına veriden elde etmesi nedeniyle, ki buna temsil öğrenmesi adı verilmektedir, son yıllarda Evrişimsel Sinir Ağları görüntü sınıflandırmada popüler hale gelmiştir. İnsan beynindeki görme merkezinin işleyişinden ilham alan bu ağlar ileri beslemeli sinir ağlarından farklı olarak konvolüsyon, havuzlama gibi işlemleri içermesinin yanında derin bir katman yapısına sahiptir. Konvolüsyon işlemi sinir ağının resmin genişlik, yükseklik ve derinlik gibi topolojik yapısını dikkate almasını sağlar. Ayrıca bu işlem, makine öğrenme sürecinin resimdeki parlama gibi küçük bozulmalara karşı daha dayanıklı olmasını ve resim içindeki objenin dönme hareketinden bağımsız hale gelmesine imkan verir. İlk konvolüsyon katmanı resim içindeki kenarları belirlerken daha derindeki katmanlar resim içindeki nesneye ait daha spesifik özellikleri ortaya çıkarırlar.

Evrişimsel Sinir Ağları özellikle 2012 yılındaki ImageNet yarışmasında yaklaşık %9'luk bir sınıflandırıcı hata oranı düşüşü sağlayan AlexNet ile dikkatleri üzerine çekmiştir. Özellikle GPU teknolojisinin gelişmesiyle de birlikte artan hesaplama gücü ile derin sinir ağları ele alınabilir hale gelmiş ve bu ağların başarısı görüntü sınıflandırma için insan gözünden daha iyi hale gelmiştir.

Evrişimsel Sinir Ağlarının öğrendiği özellikler çok yüksek doğruluk değerlerine sahip olsa da gereksiz veya ilgisiz özellikler barındırabilmektedir. Bu tezde, iyi bilinen Evrişimsel Sinir Ağları mimarileri olan AlexNet ve VGGNet'ten elde edilen özelliklere var olan çeşitli özellik seçme yöntemleri uygulanmıştır. Veriseti olarak Caltech-256 kullanılmış olup, sınıf başına 30, 60 ve %80 resim gibi üç farklı eğitim seti oluşturulmuştur. AlexNet ve VGGNet mimarileri Matlab 2018 üzerinden gerçeklenmiştir. Matlab üzerinden AlexNet, VGG16 ve VGG19 mimarilerinin tam bağlı olan FC6 ve FC7 isimli katmanlarından 4096 adet özellik çıkarılmıştır. Bu 4096 adet özelliğe, Weka yazılımı üzerinden çeşitli özellik seçme yöntemleri uygulanmıştır. Bu yöntemlerden yedi tanesi filtre yöntemi ve bir tanesi ise gömülü yöntemdir. Hesaplama maliyeti nedeniyle sarmalayıcı yöntemler kullanılmamıştır. Bu yöntemlerin sıralamış olduğu ilk 500, 1000, 2000 ve 3000 özellik kullanılarak, yine Weka üzerinden LibSVM algoritması ile sınıflandırma yapılmıştır.

Bu tezde şu ilişkiler araştırılmıştır: Eğitim setlerinin farklı olması ile özellik kalitesinin eğitim seti boyutu ile ilişkisi. Farklı katmanlarından özellikler çıkartılarak bir mimarideki katmanlar arasındaki özellik kalitesi ilişkisi. Farklı derinliklere ve hiperparametre değerlerine sahip mimariler seçilerek mimarilerin derinliği ve hiperparametre seçimlerinin özellik kalitesi arasındaki ilişki. Elde edilen sonuçlar göstermiştir ki, Evrişimsel Sinir Ağları için eğitim seti boyutunun artması, daha derin olan katmanların kullanılması ayrım gücü yüksek özellik sayısında artışa neden olmaktadır. Ayrıca bazı kombinasyonlar için, özellik seçim yöntemleri ile elde edilen daha az sayıda özellik ile yapılan sınıflandırma orjinal sayıda özellik ile yapılan sınıflandırmadan daha iyi doğruluk değeri elde etmeyi başarmıştır. Buna göre Evrişimsel Sinir Ağları mimarisinde özellik seçim algoritmalarına yer vermek doğruluk değerlerinin artmasına neden olabilmektedir. Özellikle yüksek miktarda verilerde dahi hesaplama hızı nedeniyle korelasyon tabanlı özellik seçim yöntemleri tercih edilebilir.

ACKNOWLEDGEMENTS

I am grateful to my supervisor, Assist. Prof. Dr. Buse Melis ÖZYILDIRIM, whose encouragement, guidance and support motivated me in research and study of the thesis.

It is a pleasure to thank to each and every member of the evaluation committee for their guidance.

I would like to thank my parents for their unceasing encouragement and support.

I would like to thank my children, Ali Erkin and Duru, who are my source of motivation.

I would like to thank my wife, Fatma, who never stopped supporting me, for your patience and love.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATONS

CNN	: Convolutional Neural Network
FC	: Fully Connected
FC6	: 6. Fully Connected Layer
FC7	: 7. Fully Connected Layer
GR	: Gain Ratio
IG	: Information Gain
IWSS	: Incremental Wrapper Subset Selection
PCA	: Principal Component Analysis
ReLU	: Rectified Linear Unit
SU	: Symmetrical Uncertainity
SVM	: Support Vector Machine
SGD	: Stochastic Gradient Descent
VGG	: Visual Geometry Group
VGG16	: VGG Net with 16 Layers
VGG19	: VGG Net with 19 Layers

1. INTRODUCTION

In recent years, many companies have stored a large amount of data in a systematic way whether they explicitly used them or not. In addition, on the internet, very large amounts of data are collected in various formats, such as text, pictures or video, through various sensors and mobile devices. The extraction of meaningful and useful information from this data is a problem to be addressed. The main difficulty with this problem is that the collected data contains a high level of noise. The two major causes of noise are flaws in technology that collect data and flaws in the source of data. For example, in the field of image classification, any mechanical problem in the photographic machine will be reflected as noise for subsequent processing. Most social media applications can cause pictures to be published differently than the original. Feature selection methods play an important role in removing such noisy and unnecessary features.

Moreover, the vast majority of aggregated data has more features than the number of samples. This leads to a condition called "curse of dimensionality" (Bellman, 1961) that causes deterioration in generalization ability (Donoho, 2000). According to Fukunaga (1990), the small sample size occurs when there are m samples in the n-dimensional vector space with $m <n$. Nowadays, high-dimensional low-sample size is often found in areas such as genetic microarrays, chemometry, medical imaging, text recognition, face recognition. In order to avoid this problem, feature selection methods become more of an issue. As the number of features increases and the number of samples decreases, machine learning becomes particularly difficult since the search space will take up little space and the model will not be able to correctly distinguish the data and noise (Provost 2000).

One of the main problems in machine learning is creating features that will represent the data. What has been done up to recent years in machine learning has been to draw feature vectors based on various engineering methods and field expertise from raw data such as a set of pixelated images. Such a method is both time-consuming and error-prone. Thanks to deep learning, which is a deep architecture of feedforward neural networks, this situation has given way to represent learning. Represent learning can be defined as the automatic exploration of representations needed to perform a task for a system fed by raw data. Deep learning methods transform the representation into a higher representation at every level, starting from raw input. In order to do this they use non-linear functions. By combining these transformations, features that have high discriminative power and suppress unnecessary features can be learnt (Lecun, Bengio and Hinton, 2015). For example, an image is in the form of a pixel value array, and the learned features in the first representation layer typically represent the presence or absence of edges in certain orientations and positions in the image. The second layer typically detects motifs by detecting certain edge alignments, regardless of minor changes in edge positions. The third layer may combine motifs into larger combinations corresponding to parts of known objects, and subsequent layers may perceive objects as combinations of these parts. This layered structure of deep learning leads to minimize the problems that arise from noise and position changes.

Most applications in deep learning use feedforward neural network architecture. However, in deep neural networks, the fact that the layers are completely connected causes the number of parameters to be trained to be too large and the ability to generalize to be reduced. For this reason, there are special connections between layers in deep neural networks. One of the most important of these is Convolutional Neural Networks (CNNs) (LeCun et al., 1990). CNNs have become the most widely used method of pattern recognition, especially after the success of an architecture called AlexNet (Krizhevsky, Sutskever & Hinton, 2012) in 2012 ImageNet (Deng et al., 2009) competition.

In this thesis, AlexNet and VGG16 and VGG19 (Simonyan and Zisserman, 2014) , developed by University of Oxford's Visual Geometry Group (VGG) , deep learning architectures were used. The results obtained from their fully connected sixth and seventh layers are used as a 4096-dimensional input vector for each image. After applying various feature selection methods on these 4096 dimensional vectors, the selected features are classified by Support Vector Machine (SVM). With the results obtained; it has been tried to find out whether the number of convolution layers in a CNN architecture has an effect on the quality of the feature, whether the filter size has an effect on the quality of the feature, whether there is a difference in feature quality between the first and second fully connected (FC) layers on a CNN network.

2. RELATED WORKS

Qian et al. (2016) have built an architecture based on the r * CNN (Gkioxari, Girshick and Malik, 2015) and VGG16 CNN architectures. The features of the pooling layer that they created before the first fully connected layer of this architecture were subjected to a feature selection method proposed by them. These features were obtained using the Berkeley Attributes of People dataset. Features, 500, 1500, 2500, 3500 and 4500 pieces were selected, and these features are classified with SVM.

In (Paul et al., 2016), Features were extracted from 40 tomography pictures via pre-trained MatConvNet-vgg-f, MatConv-Net-vgg-m, and MatConvNet-vgg-s CNN architectures. For this, 4096 dimensional vectors were obtained on the last fully connected layer before and after activation. Then, Relief-f and Symmetric Uncertainty feature selection algorithms were applied to these features. Selected features are classified on different classifiers.

In (Rui et al., 2016), The feature maps on the last subsampling layer of a CNN used as a feature extractor are reduced by using Linear Discriminant Analysis. The remaining features were then trained using SVM. They achieved a reduction of 20.1% in the CVC-Virtual-Pedestrian dataset, while a reduction of 9.2% in the human head dataset.

Athiwaratkun and Kang (2015) have done various experiments using 3 types of CNN architecture. In these experiments, they took features from various layers of CNN architectures, trained these features using SVM and Random Forest, and compared the results to CNN. The results showed that training with SVM and Random Forest achieved better accuracy than CNN. In another experiment, features were evaluated using Random Forest. In the case of fully connected layers, the number of important features is lower than the others.

Razavian et al. (2014) have done experiments on the features that are derived from the first fully connected layer of the OverFeat CNN architecture.

These features were trained with SVM and compared with various datasets. In a first experiment these features were subjected to size reduction up to 500 features using L2 normalization and Principal Components Analysis (PCA).

Wang et al. (2015) have proposed a feature map selection method to find discriminative feature maps and to remove unnecessary and noisy ones. In the system, they have created a new network named GNet with feature maps selected from the conv4-3 layer of the VGG16 and a small network named SNet and feature maps selected from the conv5-3 layer.

Yang et al. (2015) trained with the boosting forests of the features from the first few layers of previously trained CNN architectures such as AlexNet, VGG Net and GoogLeNet (Szegedy et al., 2014). They have also made feature selection with the method called Convolutional Channel Features.

3. MATERIAL AND METHODS

3.1. Caltech-256 Dataset

In this thesis, Caltech-256 (Griffin, Holub and Perona, 2007) dataset is used. This dataset has 256 categories of objects (Appendix D) and 1 category with background images called Clutter. There are 29780 pictures in total except for Clutter which has 827 pictures (Figure 3.1 and Figure 3.2).

Dataset	*Released*	*Categories*	*Pictures Total*	*Pictures Per Category*			
				Min	*Med*	*Mean*	*Max*
Caltech-101	2003	102	9144	31	59	90	800
Caltech-256	2006	257	30608	80	100	119	827

Figure 3.1. Caltech-256 distribution of pictures (Griffin, Holub and Perona, 2007)

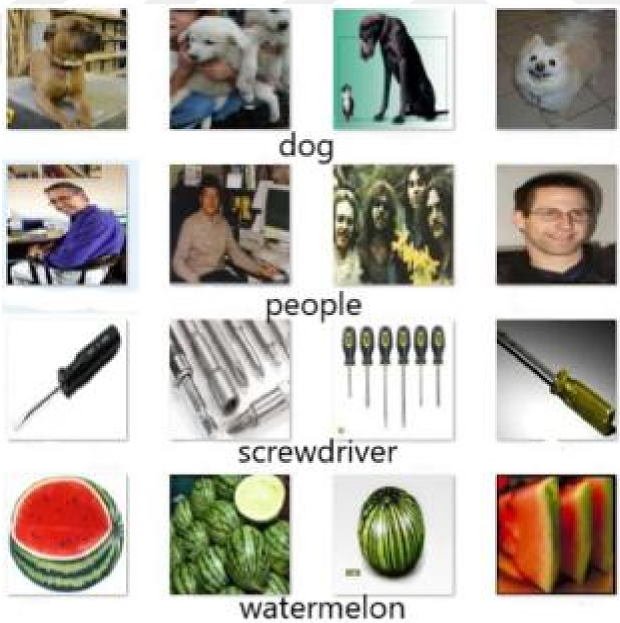


Figure 3.2. Some images from Caltech-256 dataset

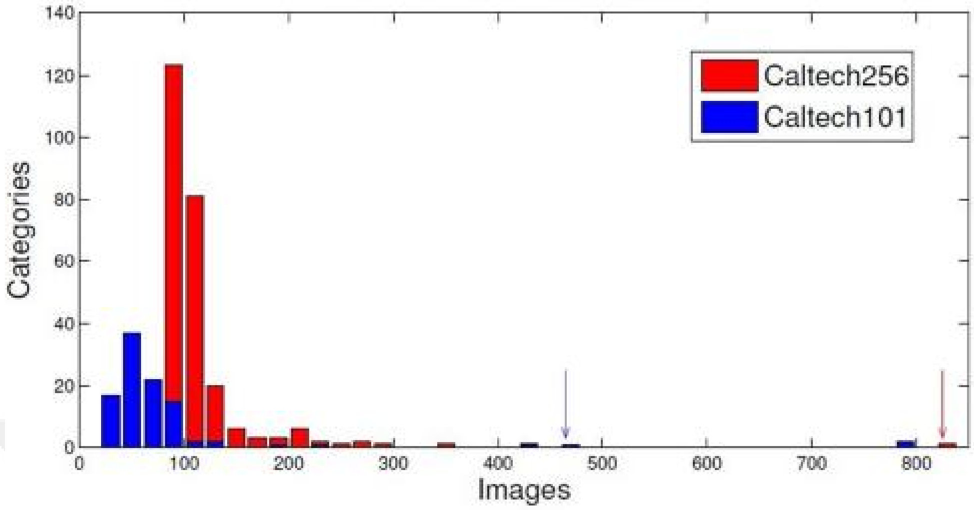


Figure 3.3. Caltech-256 distribution of category sizes (Griffin, Holub and Perona, 2007)

Figure 3.3 shows the distribution of the images of the Caltech-256 dataset according to classes. The classes that contain much more pictures than average are: airplanes 800, motorbikes 798, faces-easy 435 and t-shirt 358.

3.2. Feature Selection

Ideally, feature selection is to find a subset of features that are minimal and sufficient for the concept of the target (Kira and Rendell, 1992) . The M features are selected to be a M <N with a set of N features and the value of a criterion function is optimized over all these selected M-dimensional sub-sets (Narendra and Fukunaga, 1977).

The purpose of feature selection is to improve or not to cause a significant reduction in the classifier's predictive accuracy by using only selected features as close as possible to the original class distribution (Koller and Sahami, 1996).

The feature selection for the classification problem is intended to select a subset of features that are better discriminative than others. In other words, it is to choose features that can distinguish samples belonging to different classes. If a feature has a high correlation with a class, it is said that the feature and class are related to each other.

3.2.1. Feature Selection Steps

In general, a feature selection method involves four basic steps (Dash and Liu, 1997): subset generation procedure, subset evaluation function, stopping criteria and validation procedure.

3.2.1.1. Subset Generation

The subset generation procedure is basically a search procedure that creates the subset to be evaluated. The generation procedure may begin without any features, with all features, or with a random subset of features. In the first case, the features are added recursively, in the second case they are removed, and in the third case, they can be added or removed according to the situation (Langley, 1994).

If the original feature set contains N features, then the total number of candidate sub-sets will be 2^N. This produces a very large number even for medium-sized N. For this reason, finding an optimal subset of properties is an NP hard problem (Liu and Motoda, 2007) . There are different approaches to overcome this problem: complete, heuristic and random.

3.2.1.1.(1). Complete Search

To find the best subset, this generation procedure performs a complete search and thanks to using different heuristic functions, it evaluates fewer clusters than 2^N. Feature subset optimization is guaranteed because it uses functions that have backtracking such as branch and bound (Narendra and Fukunaga, 1977) , best first search ().

3.2.1.1.(2). Heuristic Search

At each iteration of this generation procedure, all features not yet selected or rejected are taken into account for selection or rejection. the production of sub-set is primarily monotone (increasing or decreasing). For example, greedy stepwise that is a hill climbing search is such a function.

3.2.1.1.(3). Random Search

Examples of this generation function include: Ant colony optimization, particle swarm optimization, genetic algorithm, random mutation hill-climbing and simulated annealing.

3.2.1.2. Evaluation Function

After a subset is produced by a generation procedure, it is evaluated by an evaluation function. The evaluation function is an iterative operation and the result is compared with the previous result at each operation. Each evaluation function finds its own optimal subset. Thus, an optimal subset selected using an evaluation function may not be the same as another evaluation function. Dash and Hiu (1997) divided evaluation functions into five according to various characteristics.

3.2.1.2.(1). Distance Measures

In a problem of two classes, if an X feature set creates a greater probability among the classes, it is preferred to the other Y feature set. If the difference between them is zero, then X and Y are indistinguishable.

3.2.1.2.(2). Information Measures

These measures are entropy-based measurements. The feature of which information measure is higher than another feature is preferred because the discrimination power is higher than the other one. For example, information gain, gain ratio and symmetrical uncertainty.

3.2.1.2.(3). Dependence Measures

Correlation-based measurements. It is aimed to obtain feature subset with low correlations between features each other and high correlation with class label. Example, pearson correlation coefficient.

3.2.1.2.(4). Consistency Measures

The basic idea behind these metrics is that in order to estimate the concept of the samples or the class value, only a selected set of properties and a data set must be consistent. That is, if there are two samples with different class values, the predictive feature values of these samples cannot be the same (Arauzo-Azofra, Benitez and Castro, 2008).

3.2.1.2.(5). Classifier Error Rate Measures

It is based on the idea of testing the quality of the selected subset of features with unseen samples. Although accuracy is higher than other methods, it is more costly than others (John, Kohavi and Pfleger, 1994).

The evaluation of these measures according to 3 criteria can be as follows:

1. *Generality:* While the classifier error rate is dependent on the result of a classifier, other measurements have the ability to generalize.
2. *Time complexity:* The time it takes to select a subset of features; distance, information and dependence measures are low; consistency measurement moderate; The classifier error rate is high.
3. *Accuracy:* The prediction made using the selected subset is higher than the classifier error rate.

3.2.1.3. Stopping Criteria

The absence of a stopping criteria for each of the generation procedures and evaluation functions may cause the feature selection process to take a very long time. According to generation procedures, the stopping criteria may be following: whether the predefined number of features is selected and whether a predefined number of repetitions has been reached. According to an evaluation function, the stopping criterion is: no addition or subtraction of any feature constitutes a better subset, and an optimal subset is obtained according to some evaluation functions.

3.2.1.4. Validation Procedure

The validation procedure is not part of the feature selection directly. After the selection process is over, the selected subset should be verified with a test set and the quality of the subset should be compared with the other subset.

3.2.2. Feature Selection Types

There are a number of feature selection methods divided into 3 groups based on the relation between feature selection algorithm and classifier learning algorithm: filter, wrapper, embedded (Guyon and Elisseeff, 2003).

3.2.2.1. Filter

The feature selection algorithms that adopt this method are independent of the classifier algorithm and use feature selection as a pre-process step. They have a low calculation cost and are good in terms of their generalization ability. The filter model is based on measures of general characteristics of training data such as distance, consistency, dependency, information correlation.

3.2.2.2. Wrapper

They use the learning algorithm to evaluate the predictive performance of the selected subset of features. This process provides better feature selection than the filter method but brings high computational cost.

A specific learning algorithm is used to select a subset of features in the search field, maximizing the accuracy of the learning algorithm. In other words, the wrapper model is an iterative search process in which the results of each iteration of the learning algorithm are used to guide the search process.

3.2.2.3. Embedded

They perform feature selection during the training process and are usually specific to learning algorithms. It can be seen as an approach that selects the best subset of features for the learning algorithm. Compared to Wrappers, they can catch dependencies between features with a lower computational cost.

In the embedded model, a specific learning algorithm is trained by the original feature set, and the results obtained are used to determine the suitability of each feature. In other words, the feature selection process is embedded in the training of the learning algorithm.

3.2.3. Weka Feature Selection Methods

In this thesis, Weka (Frank, Hall and Witten, 2016) is used to apply feature selection methods. The methods used are listed below:

3.2.3.1. Chi-squared Ranking Filter

This filter is based on the χ2 test statistic (Liu and Setiono, 1995). It is formulated as follows:

$$\chi 2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3.1}$$

where O_i is frequency of observations, E_i is expected frequencies.

3.2.3.2. Correlation Ranking Filter

Evaluates the rate of a feature by measuring the Pearson's correlation between it and the class (Hall, 1999).

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \tag{3.2}$$

where X is a feature, Y is a class and $\overline{X}$ is mean of X_i.

3.2.3.3. Information Gain Feature Evaluator

Information Gain (IG) is used to measure the dependency between a class value and a feature. The information gain, for X a feature and C class value, are calculated as follows:

$$IG(X, C) = H(X) - H(X|C) \tag{3.3}$$

where $H(X)$ represents entropy and is calculated as:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{3.4}$$

Conditional entropy is calculated as follows:

$$H(X|Y) = -\sum_j p(y_j) \sum_i p(x_i|y_j) \log_2 p(x_i|y_j) \tag{3.5}$$

The feature with high information gain is preferred.

3.2.3.4. Gain Ratio Ranking Filter

The Gain Ratio (GR) between an X feature and class C is calculated as follows:

$$GR(X,C) = \frac{IG(X,C)}{H(X)} \tag{3.6}$$

The feature with high gain ratio is preferred.

3.2.3.5. Symmetrical Uncertainty Ranking Filter

The Symmetrical Uncertainty (SU) between an X feature and class C is calculated as follows:

$$SU(X,C) = 2\,\frac{IG(X,C)}{H(X)+H(C)} \tag{3.7}$$

The feature with high symmetrical uncertainty is preferred.

3.2.3.6. ReliefF Ranking Filter

This algorithm introduced by Kira and Rendell (1992).

$$S_i = \frac{1}{2}\sum_{k=1}^{K} d\left(X_{ik} - X_{iM_k}\right) - d\left(X_{ik} - X_{iH_k}\right) \tag{3.8}$$

X : Instance set

M_k : on the i. feature the values of the closest instances to the x_k instance having the same class label.

H_k : on the i. feature the values of the closest instances to the x_k instance having the different class label.

$d(.)$: a distance metric.

3.2.3.7. OneR Feature Evaluator

This feature selection method evaluates and sorts each feature individually using the OneR classifier.

The pseudo code for the OneR classifier is denoted as follows:

```
for each feature
{
        for each value of that feature
        {
                compute the class distribution based on feature value
                Class_label = select most frequent class
                create a rule : feature = value = > Class_label
        }
                Calculate the error rate of the rule on the whole dataset
}
```

3.2.3.8. Incremental Wrapper Subset Selection with embedded Naive Bayes

This algorithm introduced by Bermejo, Gamez and Puerta (2014). In particular, they have developed this for high-dimensional datasets. They used the SU algorithm as a feature filter. They then tried to find the best feature subset by applying 5-fold validation. They performed an embedded feature selection using Naive Bayes classifier.

3.3. Convolutional Neural Networks

When using fully connected feedforward neural networks to classify images, too many parameters need to be trained, even for architectures with a relatively small number of hidden layers. CNNs have managed to have fewer parameters in a deeper architecture.

CNNs are inspired by the work of human visual cortex (Hubel and Wiesel, 1959). Visual information from the eyes reaches the visual cortex after passing through the various layers of the brain. The visual fields V1, V2 and V4 and the inferior temporal gyrus from these layers are those related to object recognition (Figure 3.4). While the first fields give basic information such as the edge information of the view, complex information like face is obtained in the last layer (Haxby, Hoffman and Gobbini, 2000).

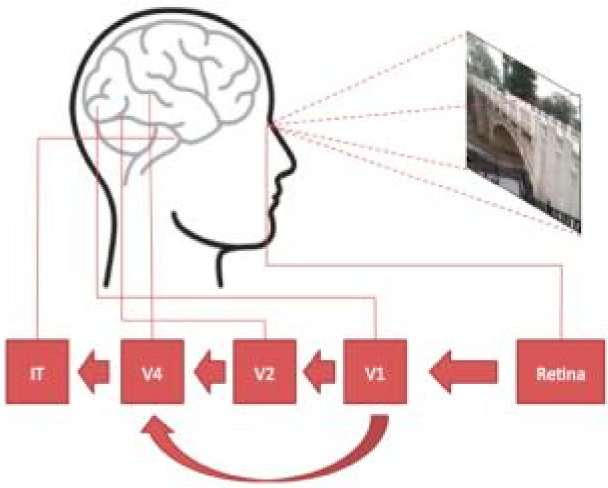


Figure 3.4. Ilustration of Visual Cortex (Wang and Raj, 2017)

To understand a CNN architecture (Figure 3.5) and to know what it is the output, we need to know the following concepts: convolution, pooling, activation and dropout.

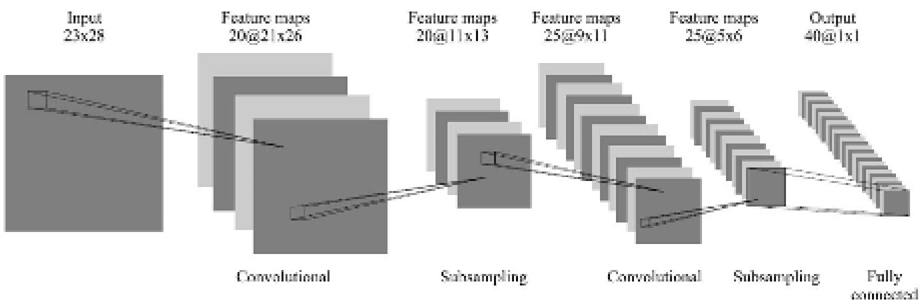


Figure 3.5. Example of basic CNN architecture (Lawrence, Giles, Tsoi and Back, 1997)

Images are constructed with channel axes in which width, height and color information are stored and stored as multidimensional arrays. Regardless of the size of a neural network, a vector is given as input, multiplied by a weight matrix, and a bias vector is added to this result. However, feedforward neural networks do not take this particular structure of images into account. Also, when FCNN receives an image as input, it is not invariant to transformation and local deformation (LeCun and Bengio, 1995). CNNs differ from feedforward neural networks because it takes into account this topological structure of the images. A convolution operation is a linear transformation that preserves the sequence structure. Because only the specific input structure affects a given output structure, it is sparse. it has weight sharing feature because the same weights are used in more than one unit at the input (Dumoulin and Visin, 2018)

The output of a convolution layer is found by convolving each convolution kernel (also known as a filter) on the picture. The output will be a picture (called feature maps) and the number of feature maps will be the same as the number of the kernel. In mathematics, the convolution operation is denoted by $*$, and for an input image I and K of size $h \times w$ convolution kernel are calculated as follows:

$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} . I_{x+i-1, y+j-1} \tag{3.9}$$

We can show the practical implementation of this mathematical notation in Figure 3.4. :

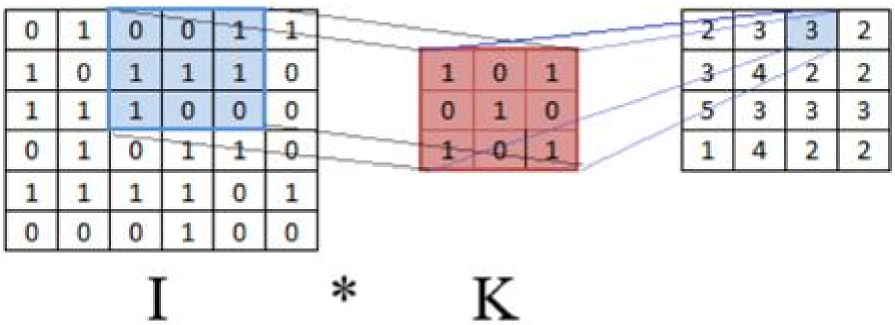


Figure 3.6. Illustration of convolution process

The amount of shifting of the kernel on an image is called ***stride***, and the value of the stride is influenced by the size of the feature map. It is also possible to add a zero to the end of each axis of a picture or a feature map. This is called *zero padding*. Zero padding also has an effect on the size of the feature map. The purposes of zero padding are to ensure that the feature map dimension remains at the dimension to be used in subsequent convolution layers and that input pixels contribute equally to the resulting feature map.

When all these concepts are taken into consideration, the size of the feature map can be calculated as follows:

$$o = \left\lfloor \frac{i+2p-k}{s} \right\rfloor + 1 \tag{3.10}$$

o = size of output feature map

i = size of input

p = size of zero padding value

k= size of kernel

s = size of stride value

3.3.1. Pooling

The main purpose of the pooling layer is to reduce the size of the feature map that occurs after performing convolution with many filters. For this reason, it is called subsampling, too. Finding feature maps that occur after a pooling process is similar to finding feature maps found at the end of a convolution process. The pooling filter of a certain size is shifted by a certain stride value and is reduced in size by a certain operation. This operation is usually performed as taking maximum. Also, average pooling and stochastic pooling (Zeiler and Fergus, 2013) operations are proposed. Scherer, Müller and Behnke (2010) showed that max pooling produces better results than average pooling.

It should be noted that the pooling process is applied separately to the resulting feature maps in the convolution process.

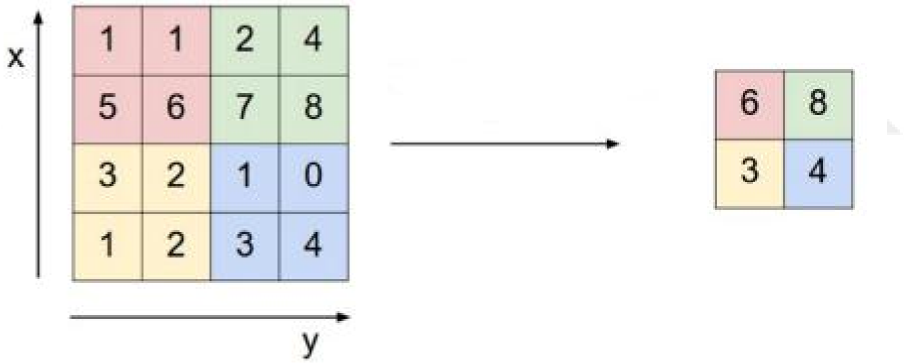


Figure 3.7. Illustration of pooling process

The size of the resulting structure in the pooling operation is similar to the size of the feature map. The only difference is that there is no zero padding in the pooling process:

$$o = \left\lfloor \frac{i-k}{s} \right\rfloor + 1 \quad (3.11)$$

3.3.2. Activation Function

After convolution, the results are usually passed through an activation function. There are various activation functions used in neural networks. The most common use of these activation functions is that they are nonlinear and differentiable. If the activation function is not linear, a neural network can learn any non-linear function with as many neurons and layers as possible. The reason that the activation functions are differentiable is that a neural network is generally trained with gradient-based optimization methods.

The most commonly used activation function in deep neural networks is the Rectified Linear Unit (ReLU). ReLU, which is formulated as (3.12) and visualized as Figure 3.8, usually provides much faster learning in deep architectures (Glorot, Bordes and Bengio, 2011).

$$ReLU = \max(0, z) \tag{3.12}$$

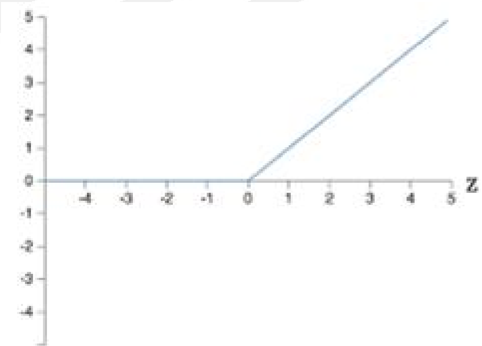


Figure 3.8. Graph of ReLU activation function

3.3.3. Fully Conntected Layer

Each neuron in a fully connected layer is connected to each neurons in previous layer by their own weights. Commonly, fully connected layers are used to enhance the classification power of features extracted from a picture by convolution layers. This is achieved by increasing the non-linearity.

3.3.4. Dropout

Dropout (Srivastava et al., 2014) is a regularization technique used to prevent overfitting. What it does is roughly prune the net. For this, a random value between 0 and 1 is generated for each neuron on the network, and neurons smaller than a predetermined threshold value are removed from the network. This layer is usually used after fully connected layers. The reason is that fully connected layers can increase non-linearity and cause the network to overfitting.

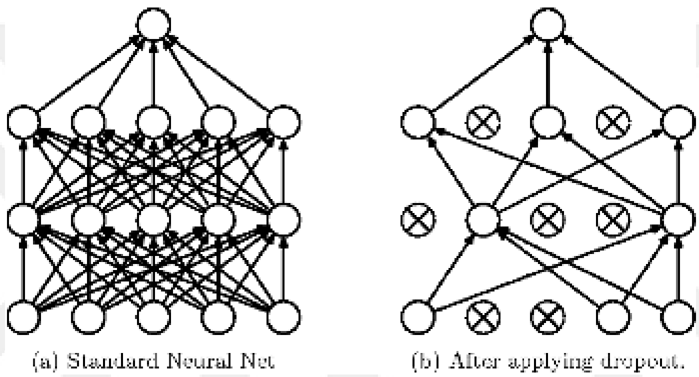


Figure 3.9. Illustration of Dropout (Srivastava et al., 2014)

3.3.5. Output Layer

The output of a CNN architecture is a layer that produces a probability value for the image given as input and contains neurons equal to the number of classes. For example, the output layer of CNNs designed for ImageNet contains 1000 neurons with a class number.

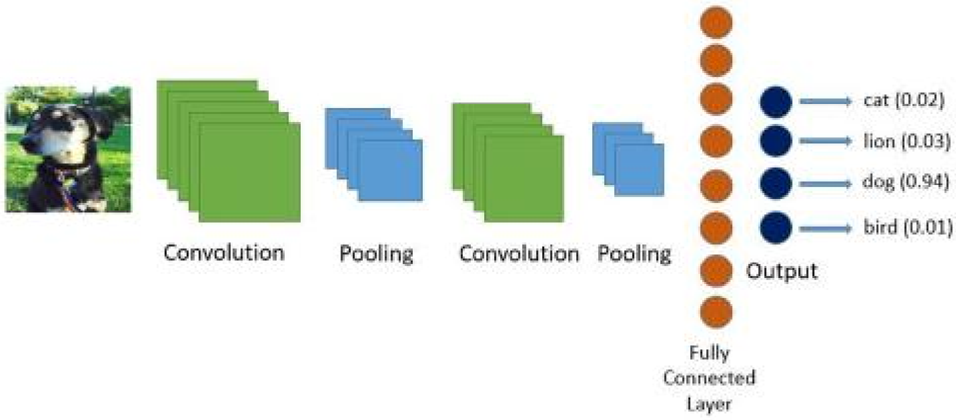


Figure 3.10. Testing a four-class CNN architecture

Figure 3.10 shows a small CNN architecture that addresses a classification problem with four classes. After CNN training, a picture of a dog in the test set that was never used during training was given as input to CNN. CNN has produced a probability value for this picture, and CNN says this is a picture of a dog because the third node of the output layer, which shows the probability value of the dog class, has the highest probability value of 0.94.

In multi-class CNN architectures, Softmax (equation 3.13) is the most commonly used method for the output layer to produce a probability value.

$$Softmax = \frac{e^{z_j}}{\sum_{n=1}^{N} e^{z_k}} \tag{3.13}$$

where N is number of classes, z is a vector of the input to the output layer.

3.3.6. CNN Architecture

In this section, two CNN architectures, AlexNet and VGG Net, which are used as feature extractors in this thesis, are introduced in detail.

3.3.6.1. AlexNet

Used in the 2012 ImageNet contest, this architecture has been a milestone for deep learning. Because the Top-5 error rate, which was 26.1% in that year's competition, has been reduced to 16.4% (with 7 convolution layers 15.3%). After this year, the best results in the ImageNet competition have always been achieved with CNNs (Figure 3.11). It was trained simultaneously on two GTX 560 GPUs for 6 days. Total number of parameters is 60.965.224.

Inputs of AlexNet are color images in the size 227x227x3 (Although it seems 224x224x3 in the original article, the calculations are true for 227x227x3).

In the first layer, there are 96 filters whose size is 11x11x3 with 4 stride value and 0 padding value. After these filters are convolved, feature maps of size 55x55x96 are formed. After the convolution layer, the ReLU activation function is applied. After ReLU, Local Response Normalization is applied. Then, max-pooling whose size is 3x3 with 2 stride value and 0 padding value is applied. After pooling, size of feature maps become 27x27x96.

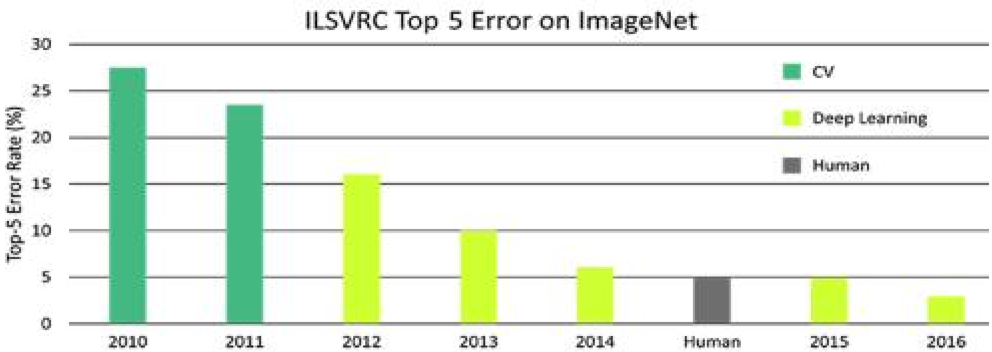


Figure 3.11. ILSVRC Top 5 .Error on ImageNet (Recker and Gribble, 2017)

In the second layer, there are 256 filters whose size is 5x5x48 with 1 stride value and 2 padding value. After these filters are convolved, feature maps of size 27x27x256 are formed. After the convolution layer, the ReLU activation function is applied. After ReLU, Local Response Normalization is applied. Then, max-pooling whose size is 3x3 with 2 stride value and 0 padding value is applied. After pooling, size of feature maps become 13x13x256.

In the third layer, there are 384 filters whose size is 3x3x256 with 1 stride value and 1 padding value. After these filters are convolved, feature maps of size 13x13x384 are formed. After the convolution layer, the ReLU activation function is applied.

In the fourth layer, there are 384 filters whose size is 3x3x192 with 1 stride value and 1 padding value. After these filters are convolved, feature maps of size 13x13x384 are formed. After the convolution layer, the ReLU activation function is applied.

In the fifth layer, there are 256 filters whose size is 3x3x192 with 1 stride value and 1 padding value. After these filters are convolved, feature maps of size 13x13x256 are formed. After the convolution layer, the ReLU activation function is applied. Then, max-pooling whose size is 3x3 with 2 stride value and 0 padding value is applied. After pooling, size of feature maps become 6x6x256.

In the sixth and seventh layer, there are two fully connected layers with 4096 neurons. After applying ReLU to these layers, a dropout was applied to reduce the number of neurons by 50%.

In the eighth and las layer, there is a fully connected layer with 1000 neurons. After this layer softmax was applied.

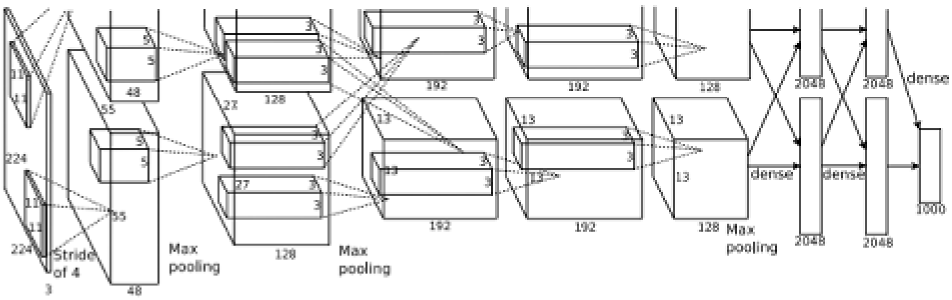


Figure 3.12. Illustration of AlexNet architecture (Krizhevsky, Sutskever and Hinton, 2012)

Parameters for training were selected as follows: Batch size is 128, Stochastic Gradient Descent (SGD) momentum is 0.9, learning rate is 0.01, L2 weight decay ise 0.0005.

3.3.6.2. VGG Net

VGG Net was introduced by The Visual Geometry Group, Oxford in 2014. At that time, the notable features are that it is deeper, and the filter size is smaller by 3x3. In the ImageNet contest in 2014, it ranks second with 7.3% top-5 error (GoogleNet with 6.7% error first). The differences from AlexNet are the use of 3x3 size filters instead of 11x11 and 7x7 and the use of repeated convolution layers before pooling.

Various architectures of VGG Net are shown in the Figure 3.13. In this thesis, 16 and 19 layered architectures, shown on columns D and E respectively, were used. The result obtained after the convolution operation with these 3x3-dimensional filters with 1 stride value and 1 padding value in both architectures was passed through the ReLU activation function. The authors reported that this case results in a nonlinear decision function with a higher discriminability than obtained with a higher-dimensional filter and three 3x3 filters sorted in succession have fewer parameters than a single 7x7 filter. In both architectures, pooling operations were done max pooling with 2x2 size filters, 2 strides and 1 padding value. A dropout of 50% was applied to the first two fully connected layers.

Total numbers of parameters are 138.355.752 for VGG16 and 143.665.448 for VGG19. Training a single net took 2–3 weeks on four NVIDIA Titan Black GPUs. Parameters for training were selected as follows: Batch size is 256, SGD momentum is 0.9, learning rate is 0.01, L2 weight decay ise 0.0005.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3.13. VGG architectures (Simonyan and Zisserman., 2014)

3.3.7. Method

The steps of the method used in this thesis are explained below.

3.3.7.1. Resizing images

The images on the Caltech-256 dataset have been resized for feature extraction from AlexNet and VGGNet architectures. The new dimensions are

227x227 for AlexNet and 224x224 for VGGNet. No cropping is applied to the images. Resizing is provided by bending.

3.3.7.2. Feature Extraction

To research feature selection methods, features extracted from AlexNet and VGGNet CNN architectures are used. Matlab 2018a has been used for feature extraction from these architectures.

For this, first, Caltech-256 dataset was divided into 30, 60 and 80% pieces for each class and the rest was the test set. Then, the images that were saved for the training set were used to extract features as 4096 dimensional matrices from the FC6 and FC7 layers of the AlexNet, VGG16, and VGG19 CNN architectures. As a result of these distinctions, a total of 18 training and test sets were obtained.

Table 3.1 shows the number of samples in the training and test sets according to the number of images used per class.

Table 3.0.1. The number of samples in the training and test sets according to the number of images used per class

Dataset	**Number of images per class**		
	30	**60**	**80%**
Training	7680	15360	23828
Test	22100	14420	5952

3.3.7.3. Feature Selection

Weka software was used to apply feature selection methods to 18 matrices with 4096 dimensions. First, these matrices are converted to ARFF files by writing a Python script. Filter-based feature selection algorithms rank 4096 features from highest to lowest. These ranked features are subdivided into sub-sets of features, with the best 500, 1000, 2000, and 3000 features. The only method that has an embedded method is to create sub-sets containing between 285 and 655 features. Thus, the total number of subsets generated is 522.

3.3.7.4. Training with LibSVM

LibSVM algorithm was applied on Weka to compare 182 feature matrices with 4096 dimensions and 522 feature matrices obtained by feature selection methods.

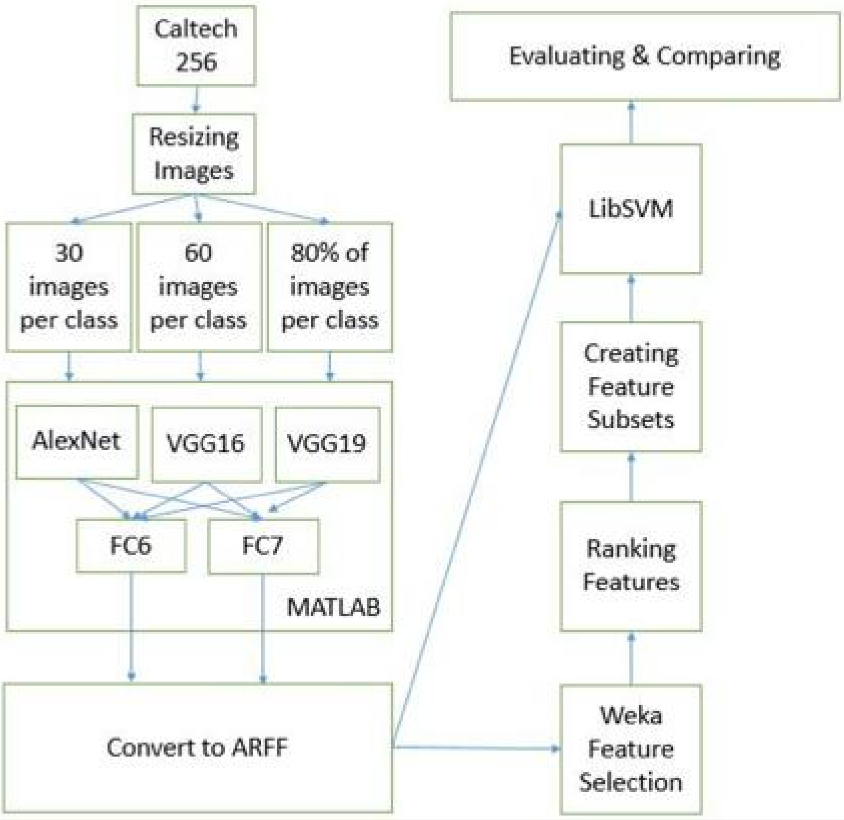


Figure 3.14. Illustration of model of thesis

4. RESULTS AND DISCUSSION

4.1. Evaluation Metrics

4.1.1. Confusion Matrix

Confusion matrix is a measure used to evaluate classification models. In a classification problem of class c, the confusion matrix is of size c x c. In this matrix, while rows represent actual labels, columns represent predicted labels. For a classification problem of two classes, the confusion matrix can be show as Figure 4.1.

		Predictive Label	
		1	-1
Actual Label	1	True Positive	False Negative
	-1	False Positive	True Negative

Figure 4.1. Confusion matrix for binary classification problem

The first row and the first column element of the Confusion matrix represents the number of instances whose actual labels are 1 and they are classified as 1. This is called True Positive (TP) instances. The first row and the second column element of the Confusion matrix represents the number of instances of which actual labels are 1 however they are classified as -1. This is called False Negative (FN) instances. The second row and the first column element of the Confusion matrix represents the number of instances whose actual labels are -1 however they are classified as 1. This is called False Positive (FP) instances. The second row and the second column element of the Confusion matrix represents the number of instances whose actual labels are -1 and they are classified as -1. This is called True Negative (TN) instances.

The Confusion Matrix provides information about how well the model is. For example, if the intersection of the first row and the second column of the matrix is equal to 25, this shows that the classifier makes mistake by classifying 25 instances belonging to class 1 as class 2.

4.1.2. Accuracy

The simplest classification measure is accuracy. It refers to the proportion of correctly classified samples in the entire test set. It is formulated as follows:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.1}$$

Accuracy metric may not produce well results in imbalanced datasets where the number of instances belonging to one class is much less than the number of instances belonging to another class.

4.1.3. Precision

Precision computes the proportion of correctly predicted positives to all predicted positives. It is formulated as follows:

$$precision = \frac{TP}{TP+FP} \tag{4.2}$$

Precision of the ith class is formulated as follows:

$$precision_i = \frac{\mathbf{M}_{ii}}{\mathbf{M}_{ii} + \sum_{j \neq i} \mathbf{M}_{ji}} = \frac{\mathbf{M}_{ii}}{\sum_j \mathbf{M}_{ji}} \tag{4.3}$$

where **M** is the confusion matrix.

If there are C classes, the whole precision of a confusion matrix can be formulated as follows:

$$precision = \sum_{i=1}^{C} precision_i \quad (4.4)$$

4.1.4. Recall

Recall computes the proportion of correctly actual positives to all actual positives. It is formulated as follows:

$$recall = \frac{TP}{TP+FN} \quad (4.5)$$

Recall of the i^th class is formulated as follows:

$$recall_i = \frac{\mathbf{M}_{ii}}{\mathbf{M}_{ii} + \sum_{j \neq i} \mathbf{M}_{ij}} = \frac{\mathbf{M}_{ii}}{\sum_j \mathbf{M}_{ij}} \quad (4.6)$$

where **M** is the confusion matrix.

If there are C classes, the whole recall of a confusion matrix can be formulated as follows:

$$recall = \sum_{i=1}^{C} recall_i \quad (4.7)$$

4.1.5. F-measure

F-measure is calculated as a harmonic mean of precision and recall metrics. Thus, it is formulated as following:

$$F - measure = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2TP}{2TP+FP+FN} \quad (4.8)$$

4.2. Performance Analysis of CNN Features without Selection

Table 4.1. shows the results of training on LibSVM (Chang and Lin, 2011) with features extracted from the FC6 and FC7 layers of AlexNet, VGG16 and VGG19 CNN architectures, 30, 60 and 80% picture numbers per class used for training dataset.

According to the results, the features obtained from the FC7 layer generally produce better results than those obtained from the FC6 layer. Also, as the number of pictures per class increases, the quality of the features also increases. For VGG architecture, the increase in the number of convolution layers gives better results in the low number of training data, while the opposite is true as the training data increases.

Table 4.1. LibSVM results of features extracted without selection

CNN	Metrics	Number of pictures per class					
		30		60		80%	
		Layer					
		FC6	FC7	FC6	FC7	FC6	FC7
AlexNet	Accuracy	61.1222	62.4706	66.5395	67.3925	68.3804	69.5901
	Precision	64.0	65.3	69.8	71.0	69.3	70.4
	Recall	61.1	62.5	66.5	67.4	68.4	69.6
	F-measure	61.6	63.0	67.0	68.1	68.1	69.3
VGG16	Accuracy	68.4796	68.7557	73.2455	73.4535	76.5289	76.4785
	Precision	71.7	71.7	77.1	76.8	77.3	77.0
	Recall	68.5	68.8	73.2	73.5	76.5	76.5
	F-measure	69.2	69.4	74.1	74.2	76.4	76.2
VGG19	Accuracy	68.8914	69.3348	73.5714	73.3495	76.4953	76.3441
	Precision	72.2	72.0	77.3	77.0	77.1	76.7
	Recall	68.9	69.3	73.6	73.3	76.5	76.3
	F-measure	69.7	69.9	74.4	74.1	76.3	76.0

4.3. Performance Analysis of IWSS with embedded Naive Bayes

Tables 4.2., 4.3. and 4.4. show the number of features selected by IWSS with embedded Naive Bayes and LibSVM results of these features extracted from AlexNet, VGG16 and VGG19 architectures of FC6 and FC7 layers with 30, 60 and 80% of images per class respectively.

Difference between LibSVM accuracy results on no filter and IWSS embedded feature selection's features are shown with Figure 4.2.

CNN	30		60		80%	
	FC6	FC7	FC6	FC7	FC6	FC7
AlexNet	(-)4.3032	(-)3.0	(-)1.9695	(-)1.1026	(-)2.2514	(-)1.2769
VGG16	(-)2.9366	(-)1.6833	(-)1.8655	(-)1.1095	(-)1.1089	(-)0.9241
VGG19	(-)2.4299	(-)1.8009	(-)1.7337	(-)0.742	(-)1.5961	(+)0.084

Figure 4.2. Difference between LibSVM results on all features and embedded feature selection's features

According to these results, feature selection on the features extracted from VGG19 FC7 layer has increased the accuracy value with only 652 features.

Table 4.2. Number of features selected by IWSS with embedded Naive Bayes and LibSVM results of these features extracted from AlexNet, VGG16 and VGG19 architectures of FC6 and FC7 layers with 30 images per class

CNN	Layer and Number of Selected Features			
	FC6	# of SF	FC7	# of SF
AlexNet	56.819	302	59.4706	285
VGG16	65.543	321	67.0724	357
VGG19	66.4615	346	67.5339	373

Table 4.3. Number of features selected by IWSS with embedded Naive Bayes and LibSVM results of these features extracted from AlexNet, VGG16 and VGG19 architectures of FC6 and FC7 layers with 60 images per class

CNN	Layer and Number of Selected Features			
	FC6	# of SF	FC7	# of SF
AlexNet	64.57	552	66.2899	455
VGG16	71.38	463	72.344	548
VGG19	71.8377	465	72.6075	546

Table 4.4. Number of features selected by IWSS with embedded Naive Bayes and LibSVM results of these features extracted from AlexNet, VGG16 and VGG19 architectures of FC6 and FC7 layers with 80% of images per class

CNN	Layer and Number of Selected Features			
	FC6	# of SF	FC7	# of SF
AlexNet	66.129	575	68.3132	536
VGG16	75.42	553	75.5544	655
VGG19	74.8992	531	76.4281	652

4.4. Performance Analysis of Feature Selection Filter Methods on 30 Pictures per Class

Table 4.5. LibSVM results of selected features extracted from AlexNet FC6 layer with 30 images per class

Filters	**500**	**1000**	**2000**	**3000**
Chi-squarred	58.543	60.1357	60.6335	60.8824
Correlation	57.1448	58.8371	60.0226	60.7421
GR	58.543	60.1357	60.6335	60.8824
IG	58.543	60.1357	60.6335	60.8824
OneR	58.4661	59.6425	60.5249	60.8009
ReliefF	56.9593	58.6787	59.9095	60.7376
SU	58.543	60.1357	60.6335	60.8824
No Filter	61.1222			

Table 4.6. LibSVM results of selected features extracted from AlexNet FC7 layer with 30 images per class

	500	**1000**	**2000**	**3000**
Chi-squarred	60.6516	61.9005	62.1312	62.3801
Correlation	60.1222	61.3620	62.1222	62.4072
GR	60.6516	61.9005	62.1312	62.3801
IG	60.6516	61.9005	62.1312	62.3801
OneR	60.3710	61.5566	62.086	62.3665
ReliefF	60.1312	61.4525	62.0769	65.4646
SU	60.6516	61.9005	62.1312	62.3801
No Filter	62.4706			

Tables 4.5 and 4.6 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the AlexNet architecture with 30 images per class, respectively.

According to the results, the feature selection methods applied to the features extracted both FC6 and FC7 layers give the same results for the Chi-squared, GR, IG and SU even 500 feature counts. This means that AlexNet is not able to produce enough distinctive features in a training set with 30 images per class. For both layers, the accuracy increases as the number of features increases. The best result for the FC6 layer is (-)0.2398 accuracy for 3000 features, while this value for the FC7 layer is (-)0.0634 for 3000 features. These results are also illustrated as Figure 4.3. and Figure 4.4.

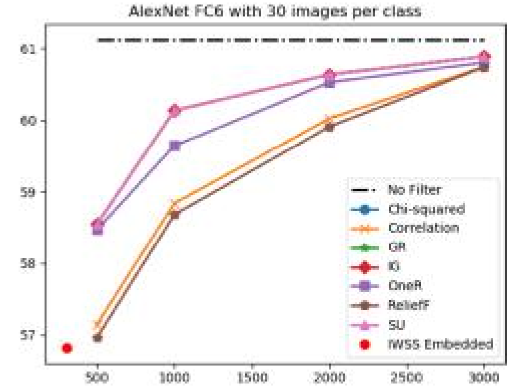


Figure 4.3. Illustration of LibSVM results obtained features extracted AlexNet FC6 layer with 30 images per class

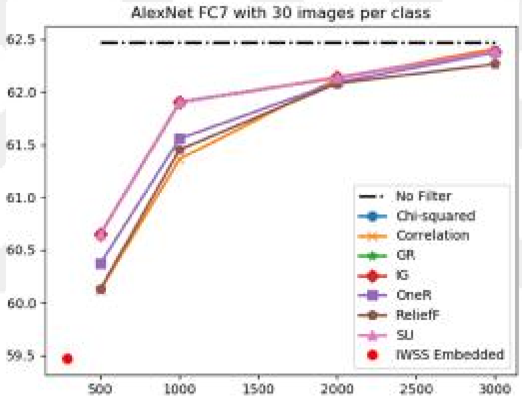


Figure 4.4. Illustration of LibSVM results obtained features extracted AlexNet FC7 layer with 30 images per class

Table 4.7. LibSVM results of selected features extracted from VGG16 FC6 layer with 30 images per class

Filters	500	1000	2000	3000
Chi-squarred	66.0	67.5249	68.1855	68.2398
Correlation	65.638	67.0679	68.1267	68.3439
GR	65.914	67.5249	68.1855	68.2398
IG	66.1493	67.5249	68.1855	68.2398
OneR	66.5792	67.629	68.1946	68.3258
ReliefF	65.086	66.3665	67.6923	68.2624
SU	65.9729	67.5249	68.1855	68.2398
No Filter	68.4796			

Table 4.8. LibSVM results of selected features extracted from VGG16 FC7 layer with 30 images per class

	500	1000	2000	3000
Chi-squarred	66.8869	68.1403	68.6018	68.7059
Correlation	66.5837	67.8552	68.4253	68.6787
GR	66.7195	68.009	68.6018	68.7059
IG	66.8462	68.1357	68.6018	68.7059
OneR	67.4977	68.2489	68.81	68.6787
ReliefF	66.3167	67.4796	68.2081	68.457
SU	67.0	68.0	68.6018	68.7059
No Filter	68.7557			

Tables 4.7 and 4.8 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the VGG16 architecture with 30 images per class, respectively.

According to these results, the features obtained from VGG16 for 30 images per class start to give the same results with the number of 1000 feature for FC6 layer and 2000 feature for FC7 layer when Chi-squared, GR, IG and SU filters are applied. This means that there is an increase in the number of distinctive features compared to AlexNet. The best result for the FC6 layer is (-)0.1357 accuracy for 3000 features, while this value for the FC7 layer is (+)0.0543 for 2000 features. Although the number of features drops from 4096 to 2000, there is an increase in accuracy. These results are also illustrated as Figure 4.5. and Figure 4.6.

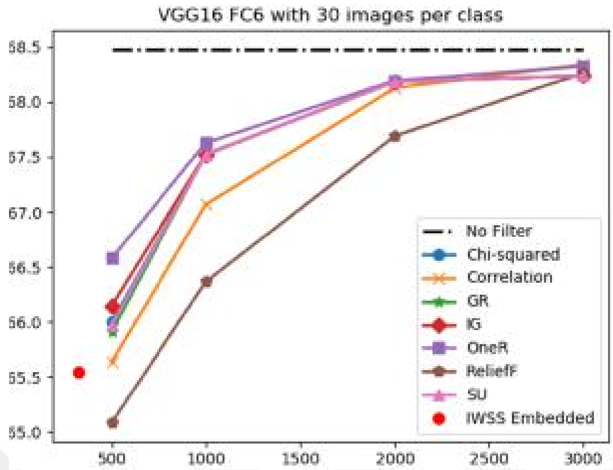


Figure 4.5. Illustration of LibSVM results obtained features extracted VGG16 FC6 layer with 30 images per class

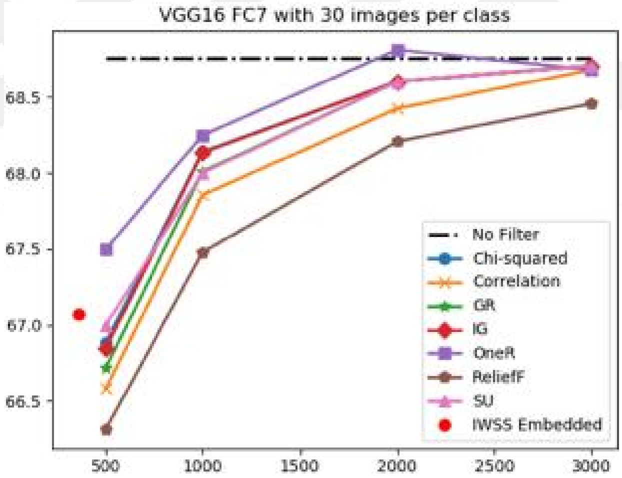


Figure 4.6. Illustration of LibSVM results obtained features extracted VGG16 FC7 layer with 30 images per class

Table 4.9. LibSVM results of selected features extracted from VGG19 FC6 layer with 30 images per class

Filters	500	1000	2000	3000
Chi-squared	66.1086	67.5611	68.4434	68.9231
Correlation	65.9593	67.4525	68.3665	68.6335
GR	66.1086	67.5611	68.4434	68.9231
IG	66.1086	67.5611	68.4434	68.9231
OneR	66.8824	67.9186	68.4751	68.81
ReliefF	65.2941	66.914	68.19	68.4706
SU	66.1086	67.5611	68.4434	68.9231
No Filter	68.8914			

Table 4.10. LibSVM results of selected features extracted from VGG19 FC7 layer with 30 images per class

	500	1000	2000	3000
Chi-squared	67.7466	68.724	69.1357	69.267
Correlation	67.362	68.2398	69.0	69.2172
GR	67.543	68.724	69.1357	69.267
IG	67.5249	68.724	69.1357	69.267
OneR	67.8552	68.6018	69.0226	69.2172
ReliefF	66.7828	68.0452	68.6244	69.0181
SU	67.6787	68.724	69.1357	69.267
No Filter	69.3348			

Tables 4.9 and 4.10 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the VGG19 architecture with 30 images per class, respectively.

The best result for the FC6 layer is (+)0.0317 accuracy for 3000 features, while this value for the FC7 layer is (-)0.0678 for 3000 features. Feature selection on the features extracted from FC6 layer has increased the accuracy value. These results are also illustrated as Figure 4.7. and Figure 4.8.

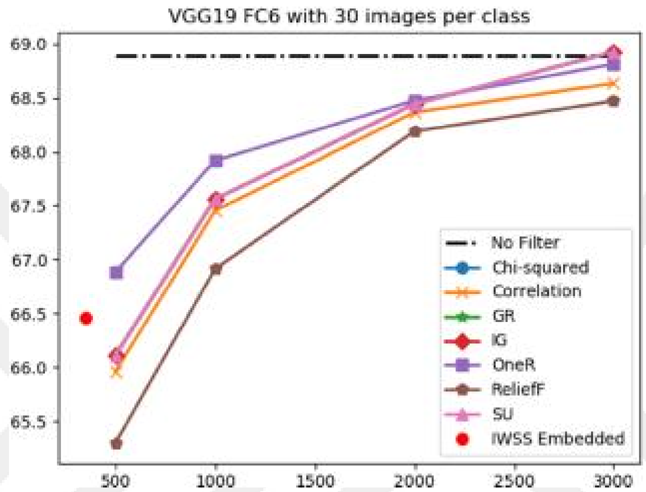


Figure 4.7. Illustration of LibSVM results obtained features extracted VGG19 FC6 layer with 30 images per class

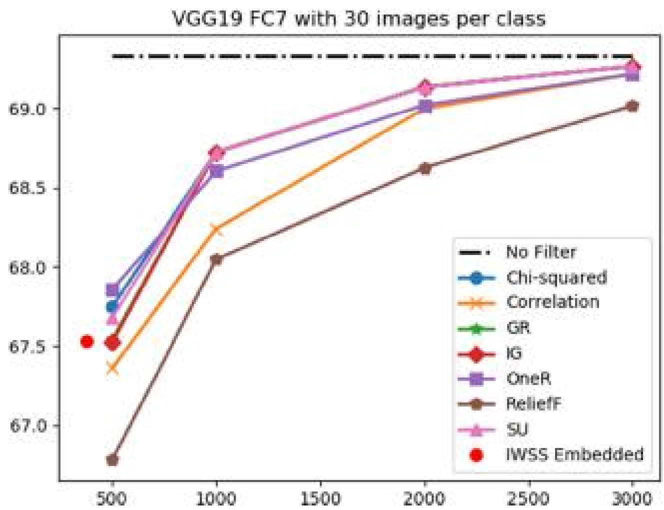


Figure 4.8. Illustration of LibSVM results obtained features extracted VGG19 FC7 layer with 30 images per class

4.5. Performance Analysis of Feature Selection Filter Methods on 60 Pictures per Class

Table 4.11. LibSVM results of selected features extracted from AlexNet FC6 layer with 60 images per class

Filters	**500**	**1000**	**2000**	**3000**
Chi-squarred	63.7101	65.0277	66.0402	66.3245
Correlation	63.5506	65.3121	65.9639	66.3523
GR	63.5645	65.1872	65.9293	66.3245
IG	63.6061	65.2705	65.8946	66.3245
OneR	64.3412	65.527	66.0402	66.5187
ReliefF	63.4951	64.6602	65.7143	66.1304
SU	63.5576	65.3259	66.172	66.3245
No Filter	66.5395			

Table 4.12. LibSVM results of selected features extracted from AlexNet FC7 layer with 60 images per class

	500	**1000**	**2000**	**3000**
Chi-squarred	65.3537	66.4147	66.9972	67.4563
Correlation	65.5617	66.6436	66.8585	67.3717
GR	65.4716	66.387	66.8724	67.2885
IG	65.6172	66.4702	67.025	67.1429
OneR	65.7906	66.6436	67.1429	67.4411
ReliefF	65.4646	66.3939	66.914	67.2954
SU	65.5548	66.3107	67.018	67.1983
No Filter	67.3925			

Tables 4.11 and 4.12 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the AlexNet architecture with 60 images per class, respectively.

According to these results, the features extracted from AlexNet for 60 images per class has more distinctive features than for 30 images per class, since for FC7 layer, all feature selection methods yields different results. This shows that as the number of training samples per class increases, the quality of the features also increases. For both layers, the accuracy increases as the number of features increases. The best result for the FC6 layer is (-)0.0208 accuracy for 3000 features, while this value for the FC7 layer is (+)0.0638 for 3000 features. Feature selection on the features extracted from FC7 layer has increased the accuracy value. These results are also illustrated as Figure 4.9. and Figure 4.10.

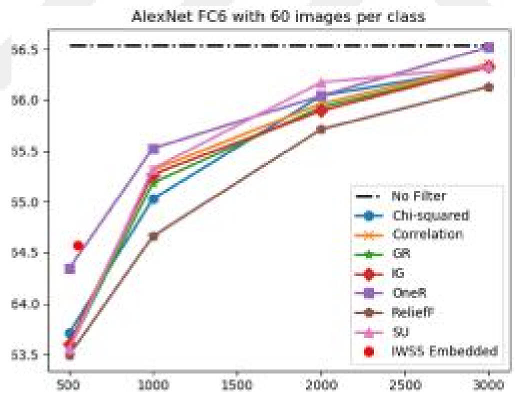


Figure 4.9. Illustration of LibSVM results obtained features extracted AlexNet FC6 layer with 60 images per class

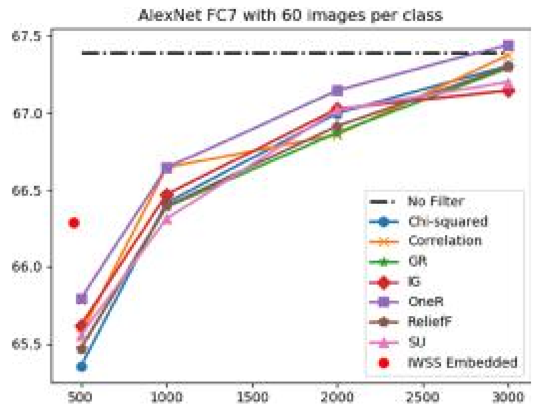


Figure 4.10. Illustration of LibSVM results obtained features extracted AlexNet FC7 layer with 60 images per class

Table 4.13. LibSVM results of selected features extracted from VGG16 FC6 layer with 60 images per class

Filters	500	1000	2000	3000
Chi-squarred	70.7143	71.9764	72.5173	72.975
Correlation	70.6241	71.9209	72.6006	72.975
GR	70.7975	72.2261	72.5659	72.9057
IG	70.8322	71.8377	72.552	73.0236
OneR	70.9778	71.9972	72.5936	72.9958
ReliefF	70.0277	71.5673	72.4272	72.8918
SU	70.8807	71.8655	72.6283	73.0444
No Filter	73.2455			

Table 4.14. LibSVM results of selected features extracted from VGG16 FC7 layer with 60 images per class

	500	1000	2000	3000
Chi-squarred	71.6019	72.5589	73.19	73.3703
Correlation	71.6089	72.8918	73.1553	73.4327
GR	71.4979	72.5381	73.0721	73.4119
IG	71.6852	72.663	73.1484	73.3634
OneR	72.1914	72.9334	73.2316	73.4743
ReliefF	71.5187	72.3925	72.9473	73.0999
SU	71.6574	72.5659	73.1831	73.3356
No Filter	73.4535			

Tables 4.13 and 4.14 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the VGG16 architecture with 60 images per class, respectively.

According to these results, the features extracted from VGG16 for 60 images per class has more distinctive features than AlexNet, since all feature selection methods, even on FC6 layer also, yields different results. For both layers, the accuracy increases as the number of features increases. The best result for the FC6 layer is (-)0.2011 accuracy for 3000 features, while this value for the FC7 layer is (+)0.0208 for 3000 features. Feature selection on the features extracted from FC7 layer has increased the accuracy value. These results are also illustrated as Figure 4.11. and Figure 4.12.

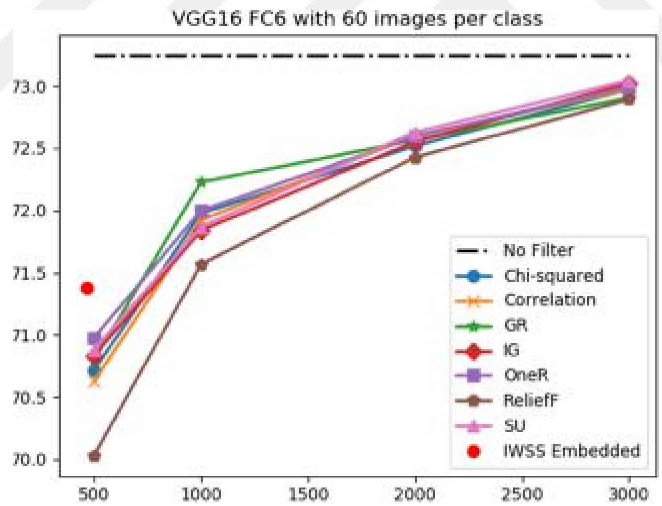


Figure 4.11. Illustration of LibSVM results obtained features extracted VGG16 FC6 layer with 60 images per class

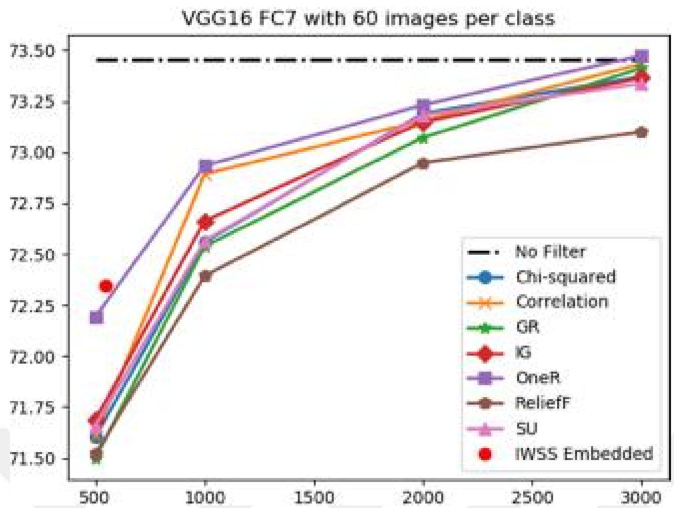


Figure 4.12. Illustration of LibSVM results obtained features extracted VGG16 FC7 layer with 60 images per class

Table 4.15. LibSVM results of selected features extracted from VGG19 FC6 layer with 60 images per class

Filters	**500**	**1000**	**2000**	**3000**
Chi-squarred	71.1096	72.1567	72.9265	73.2663
Correlation	70.9778	72.0666	72.767	73.3426
GR	70.957	72.0666	73.0374	73.2663
IG	70.8183	72.1775	72.9265	73.301
OneR	71.6852	72.448	72.9612	73.2108
ReliefF	70.3467	71.5534	72.4619	73.0929
SU	70.9015	72.0943	72.8779	73.301
No Filter	73.5714			

Table 4.16. LibSVM results of selected features extracted from VGG19 FC7 layer with 60 images per class

	500	1000	2000	3000
Chi-squarred	71.491	72.2677	73.1692	73.0791
Correlation	71.7892	72.4549	72.9542	73.2108
GR	71.3454	72.2469	73.1761	73.0236
IG	71.6436	72.1983	72.975	73.0721
OneR	71.7614	72.6838	73.0791	73.1831
ReliefF	71.81	72.3994	72.975	73.3842
SU	71.6436	72.2885	73.0513	73.0652
No Filter	73.3495			

Tables 4.15 and 4.16 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the VGG19 architecture with 60 images per class, respectively.

According to these results, the features extracted from VGG19 for 60 images per class has more distinctive features than AlexNet, too. For GR in the FC7 layer, the number of features increases from 2000 to 3000, while the accuracy decreases. The best result for the FC6 layer is (-)0.2288 accuracy for 3000 features, while this value for the FC7 layer is (+)0.0347 for 3000 features. Feature selection on the features extracted from FC7 layer has increased the accuracy value. These results are also illustrated as Figure 4.13. and Figure 4.14.

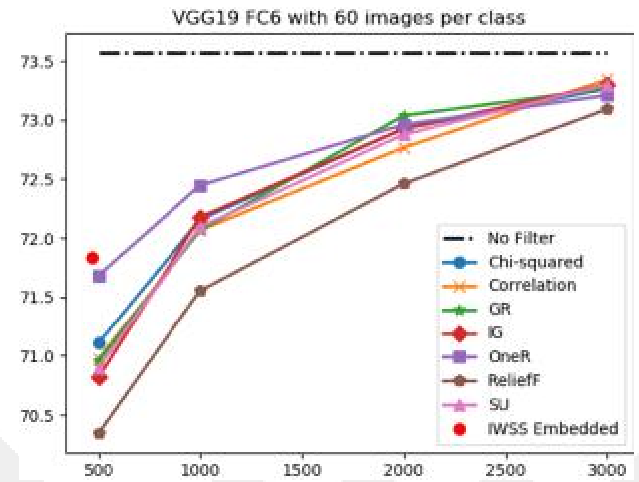


Figure 4.13. Illustration of LibSVM results obtained features extracted VGG19 FC6 layer with 60 images per class

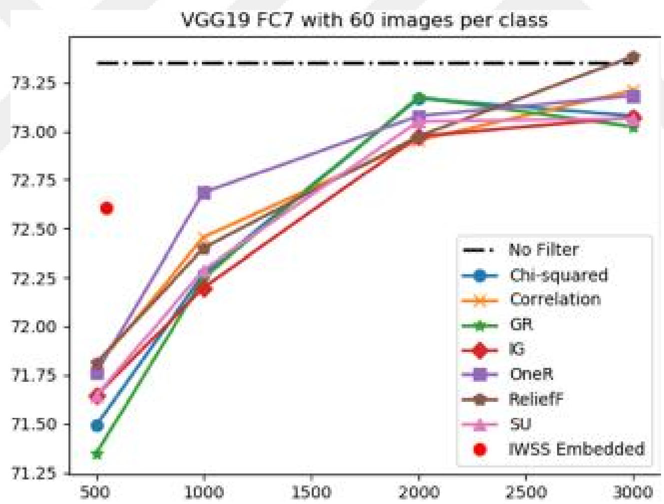


Figure 4.14. Illustration of LibSVM results obtained features extracted VGG19 FC7 layer with 60 images per class

4.6. Performance Analysis of Feature Selection Filter Methods on 80% of Pictures per Class

Table 4.17. LibSVM results of selected features extracted from AlexNet FC6 layer with 80% of images per class

Filters	500	1000	2000	3000
Chi-squarred	65.5242	67.3051	67.8931	68.5988
Correlation	65.2386	67.5907	68.2628	68.4644
GR	65.6922	67.2211	67.8763	68.582
IG	65.8602	67.3051	68.0108	68.6156
OneR	65.9274	67.9772	68.3804	68.4812
ReliefF	65.2554	66.5995	68.1284	68.1956
SU	65.9442	67.3387	67.9435	68.6492
No Filter	68.3804			

Table 4.18. LibSVM results of selected features extracted from AlexNet FC7 layer with 80% of images per class

	500	1000	2000	3000
Chi-squarred	67.4563	68.6996	69.17	69.5565
Correlation	68.078	68.6324	69.3716	69.6237
GR	67.7083	68.666	69.2036	69.5565
IG	67.6747	68.582	69.2204	69.506
OneR	67.8091	68.834	69.3884	69.6405
ReliefF	68.1788	69.002	69.4724	69.2204
SU	67.6579	68.6828	69.1868	69.4556

Tables 4.17 and 4.18 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the AlexNet architecture with 80% of images per class, respectively.

According to these results, the features extracted from AlexNet FC6 layer for 80% of images per class increased accuracy on all selection methods except ReliefF. For both layers, the accuracy increases as the number of features increases. The best result for the FC6 layer is (+)0.2688 accuracy for 3000 features, while this value for the FC7 layer is (+)0.0504 for 3000 features. Feature selection on the features extracted from FC6 and FC7 layers have increased the accuracy value. These results are also illustrated as Figure 4.15. and Figure 4.16.

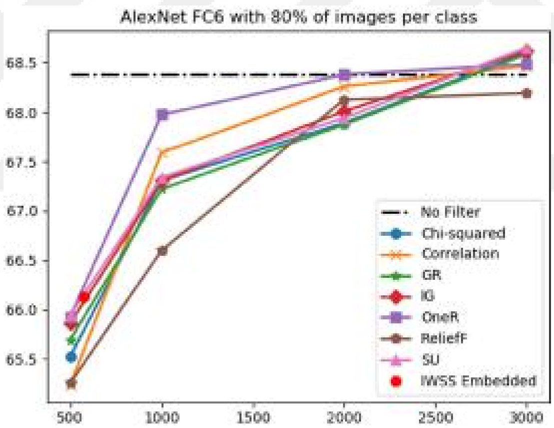


Figure 4.15. Illustration of LibSVM results obtained features extracted AlexNet FC6 layer with 80% of images per class

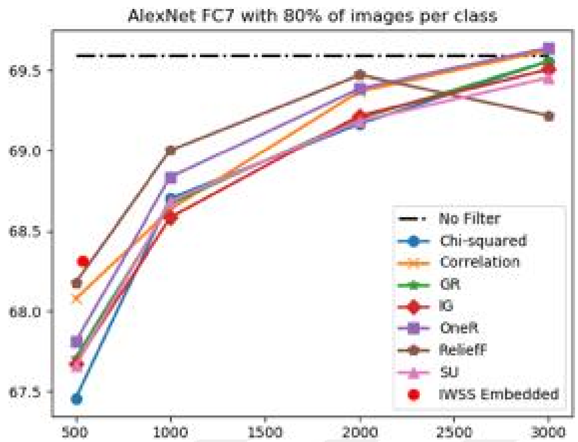


Figure 4.16. Illustration of LibSVM results obtained features extracted AlexNet FC7 layer with 80% of images per class

Table 4.19. LibSVM results of selected features extracted from VGG16 FC6 layer with 80% of images per class

Filters	**500**	**1000**	**2000**	**3000**
Chi-squarred	74.5464	75.4032	76.1425	76.6297
Correlation	74.5128	75.2016	76.0585	76.4953
GR	74.1431	75.4872	76.2097	76.6297
IG	74.4456	75.5376	76.1089	76.5625
OneR	74.7984	75.3024	76.0921	76.5289
ReliefF	74.1767	74.9328	76.0081	76.3273
SU	74.5464	75.5376	76.0417	76.5625
No Filter	76.5289			

Table 4.20. LibSVM results of selected features extracted from VGG16 FC7 layer with 80% of images per class

	500	**1000**	**2000**	**3000**
Chi-squarred	75.252	75.6888	76.1257	76.2433
Correlation	75.1176	75.5208	76.0753	76.3105
GR	74.2608	75.3864	76.0417	76.1593
IG	75.3528	75.4872	75.8905	76.3777
OneR	75.1512	75.4536	76.0249	76.1089
ReliefF	74.4624	75.2016	75.756	76.1593
SU	75.3024	75.5712	75.8905	76.3105
No Filter	76.4785			

Tables 4.19 and 4.20 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the VGG16 architecture with 80% of images per class, respectively.

According to these results, the best result for the FC6 layer is +0.1008 accuracy for 3000 features, while this value for the FC7 layer is -0.1008 for 3000 features. Feature selection on the features extracted from FC6 layer has increased the accuracy value. These results are also illustrated as Figure 4.17. and Figure 4.18.

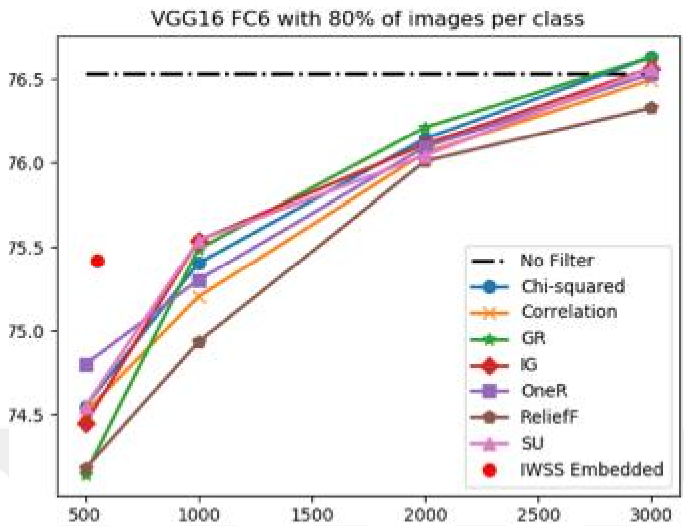


Figure 4.17. Illustration of LibSVM results obtained features extracted VGG16 FC6 layer with 80% of images per class

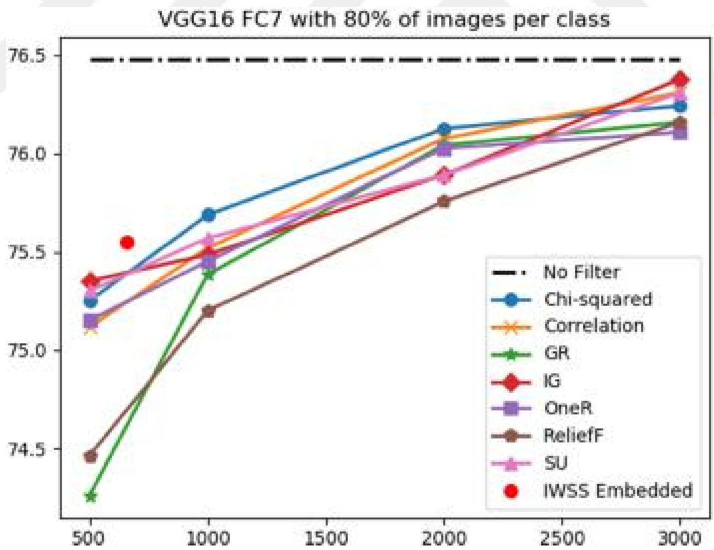


Figure 4.18. Illustration of LibSVM results obtained features extracted VGG16 FC7 layer with 80% of images per class

Table 4.21. LibSVM results of selected features extracted from VGG19 FC6 layer with 80% of images per class

Filters	500	1000	2000	3000
Chi-squarred	74.8488	75.5712	76.3105	76.5961
Correlation	74.6304	75.6552	76.0921	76.4449
GR	74.916	75.5208	76.3945	76.5793
IG	74.7984	75.6384	76.1929	76.5625
OneR	75.168	75.9745	76.6969	76.7137
ReliefF	74.2103	75.5712	76.1593	76.5121
SU	74.748	75.42	76.2097	76.3945
No Filter	76.4953			

Table 4.22. LibSVM results of selected features extracted from VGG19 FC7 layer with 80% of images per class

	500	1000	2000	3000
Chi-squarred	75.084	75.4704	76.4113	76.5625
Correlation	75.1512	76.0585	76.0753	76.4953
GR	75.1512	75.756	76.4113	76.6297
IG	75.0168	75.8233	76.4617	76.5457
OneR	75.756	75.9409	76.5289	76.5961
ReliefF	75.084	75.7056	76.2937	76.2769
SU	74.9496	75.672	76.4281	76.5121
No Filter	76.3441			

Tables 4.21 and 4.22 show the LibSVM training results on the first 500, 1000, 2000, and 3000 features obtained from feature selection methods on features extracted from the layers of FC6 and FC7 of the VGG19 architecture with 80% of images per class, respectively.

According to these results, the best result for the FC6 layer is (+)0.2184 accuracy for 3000 features, while this value for the FC7 layer is (+)0.2856 for 3000 features. Feature selection on the features extracted from FC6 and FC7 layers have increased the accuracy value. These results are also illustrated as Figure 4.19. and Figure 4.20.

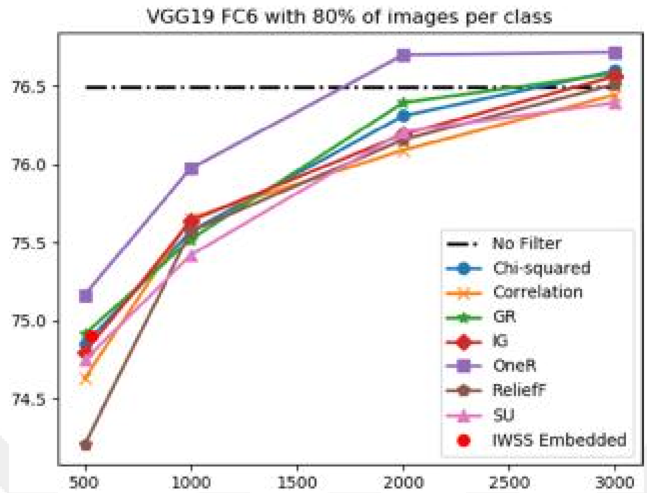


Figure 4.19. Illustration of LibSVM results obtained features extracted VGG19 FC6 layer with 80% of images per class

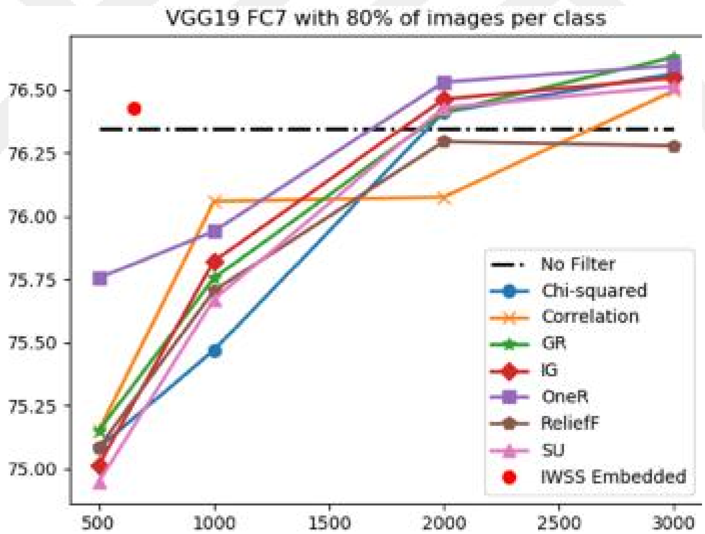


Figure 4.20. Illustration of LibSVM results obtained features extracted VGG19 FC7 layer with 80% of images per class

4.7. Performance Analysis of Feature Selection Methods by Execution Time

Table 4.23. Execution time feature selection methods used in thesis

Feature Selection Method	Number of Images per Class		
	30	60	80%
Chi-squared	00:00:47	00:02:52	00:05:47
Correlation	00:00:15	00:00:37	00:01:04
GR	00:00:46	00:02:53	00:06:14
IG	00:00:44	00:02:55	00:06:12
OneR	00:10:58	00:21:12	00:35:52
ReliefF	01:27:24	04:28:58	05:55:52
SU	00:00:47	00:02:49	00:05:53
IWSS with embedded Naive Bayes	00:25:44	00:57:39	01:05:43

Table 4.24. Time taken to build model of LibSVM (seconds)

Number of Selected Features	Time taken to build model		
	30	60	80%
500	37.26	100.96	271.33
1000	81.57	284.13	699.79
2000	190.92	677.69	1020.99
3000	243.67	920.80	1774.51
No Selection	376.15	1067.85	2441.25

Tables 4.23. and 4.24. show the execution time of feature selection methods and time taken to build model of LibSVM, respectively. These results obtained with following system: Intel® Core™ i7-7700 CPU @ 3.60Ghz, 16 GB RAM and HDD disk.

According to Table 4.23. results, correlation is the fastest among all methods.

According to Table 4.24. results, a 2-fold increase in the number of samples per class in the training set causes a 3-fold increase in LibSVM training time, approximately.

4.8. Previous Studies using Caltech-256 Dataset

Mahmood et al. (2016) extracted the features from the last residual unit of convolutional layers three, four and five of ResNet (He et al., 2015) 50 and 152 layers. Each feature vector extracted from these layers is referred to as ResFeats-50 and ResFeats-152. They used LibSVM as classfication method.

Zeiler and Fergus (2014) won first place in the ILSVRC 2013 competition by making changes in the hyperparameters of AlexNet architecture and downsizing in stride and filter sizes. Architecture is known as ZF Net.

Zheng et al. (2016) have worked on AlexNet and VGG Net with the changes to the image resizing method and the pooling process in the lower layers.

Table 4.25. Results of previous works used Caltech 256

Method	30	60	80%
ZF Net	70.6	74.2	-
ResFeats-50	75.4	79.3	-
ResFeats-152	78	81.9	-
Zheng et al.'s AlexNet	-	74.3	-
Zheng et al.'s VGG Net	-	86	-
Sohn et al. (2011)	42.05	47.94	-
In this thesis AlexNet	62.4706	67.4411	69.6405
In this thesis VGG16	68.7557	73.4743	76.6297
In this thesis VGG19	69.3348	73.5714	76.7137

The results of these and some of the other studies and in this thesis best results are shown in the Table 4.25.

The best results of AlexNet are obtained by using 4096 features extracted FC7 layer for 30 images per class, 3000 features selected by OneR filter among features extracted FC7 layer for 60 images per class and %80 of images per class.

The best results of VGG16 are obtained by using 4096 features extracted FC7 layer for 30 images per class, 3000 features selected by OneR among features extracted FC7 layer for 60 images per class and 3000 features selected by Chi-squarred and GR filters among features extracted FC6 layer for 80% of images per class.

The best results of VGG19 are obtained by using 4096 features extracted FC7 layer for 30 images per class, 4096 features extracted FC6 layer for 60 images per class and 3000 features selected by OneR filter among features extracted FC6 layer for 80% of images per class.

5. CONCLUSIONS

In this study, the effect of the feature selection methods on CNN is examined. Feature selection methods have been applied to features extracted from the fully connected layers of the well-known CNN architectures AlexNet, VGG16 and VGG19. The effect of architectures' depth, selecting different fully connected layer, and number of training data can be explained as follows:

Given the depth difference between AlexNet and VGG architectures, feature selection methods based on the features of the VGG architecture, a deeper architecture, give more effective results. The lack of distinctive features for AlexNet architecture, especially in the case of low number of training data, has led many filters to produce the same results. It has also been observed that the VGG19 architecture produces more effective results than the VGG16 architecture, especially as the training data increase. Taken all this into account, it can be said that the increase in the depth of CNN architecture caused the feature selection methods to give effective results.

It has been observed that feature selection methods performed on features extracted from the FC7 layer select features with higher accuracy values than the FC6 layer. It can be said that the FC7 layer features are more selective than the FC6 layer features.

It has also been observed that the increase in the training data plays a role in the way feature selection methods give effective results. Taking all this into account, it is considered appropriate to incorporate a feature selection method into a CNN architecture. For this, correlation-based methods can be chosen which produce faster results than others. Thus, by reducing both the number of parameters, the training time can be shortened, and the accuracy value can be increased.

REFERENCES

Arauzo-Azofra, A., Benitez, J.M. and Castro, J.L., 2008. Consistency measures for feature selection, Intelligent Information Systems, 30: 273-292

Athiwaratkun, B., Kang, K., 2015. Feature Representation in Convolutional Neural Networks, arXiv preprint, arXiv:1507.02313.

Bellman, R. E., 1961. Adaptive Control Processes: A Guided Tour, Princeton University Press, volume 4.

Bermejo, P., Gamez, J. A., and Puerta, J. M., 2014. Speeding up incremental wrapper feature subset selection with Naive Bayes classifier, Knowledge-Based Systems, 55:140-147.

Bo, L., Ren, X., and Fox, D., 2013. Multipath sparse coding using hierarchical matching pursuit, CVPR.

Chang, C.-C., and Lin, C.-J., 2011. LIBSVM : a library for support vector machines, ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27.

Dash M., and Liu, H., 1997. Feature selection for classification, Intelligent data analysis, 1(1-4):131–156.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, In CVPR.

Donoho, D. L., 2000. High-dimensional data analysis: The curses and blessings of dimensionality, AMS Math Challenges Lecture, pages 1–32.

Dumoulin V., and Visin, F., 2018. A guide to convolution arithmetic for deep learning, arXiv preprint, arXiv:1603.07285v2.

Frank, E., Hall, M. A. and Witten, I. H., 2016. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, Academic Press, New York, 579p.

Gkioxari, G., Girshick, R., and Malik, J., 2015. Contextual action recognition with r*cnn, IEEE International Conference on Computer Vision (ICCV), 1080–1088.

Glorot, X., Bordes, A., and Bengio, Y., 2011. Deep sparse rectifier neural networks, 14th International Conference on Artificial Intelligence and Statistics, 315–323.

Griffin, G., Holub, A., and Perona, P., 2007. Caltech-256 Object Category Dataset. California Institute of Technology

Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection, The Journal of Machine Learning Research, 3:1157–1182.

Hall, M. A., 1999. Correlation-based Feature Selection for Machine Learning, PhD thesis, Department of Computer Science, University of Waikato.

Haxby, J. V., Hoffman, E. A. and Gobbini, M. I., 2000. The distributed human neural system for face perception, Trends in cognitive sciences, 4(6):223-233.

He, K., Zhang, X., Ren, S., and Sun, J., 2015. Deep Residual Learning for Image Recognition, arXiv preprint, arXiv:1512.03385

Hubel, D. H., Wiesel, T. N., 1959. Receptive Fields of single neurones in the cat's striate cortex, The Journal of physiology, 148(3):574-591.

John, G.H., Kohavi, R. and Pfleger, K., 1994. Irrelevant features and the subset selection problem, Proceedings of the Eleventh International Conference on Machine Learning, 121–129.

Kira, K., and Rendell, L.A., 1992. The feature selection problem: Traditional methods and a new algorithm, Proceedings of Ninth National Conference on Artificial Intelligence, 129–134.

Koller, D., and Sahami, M., 1996. Toward optimal feature selection, Proceedings of International Conference on Machine Learning.

Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. Image Net Classification with Deep Convolutional Neural Network, NIPS.

Langley, P., 1994. Selection of relevant features in machine learning, Proceedings of the AAAI Fall Symposium on Relevance, 1–5.

Lawrence, S., Giles, C. L., Tsoi A. C. and Back, A. D., 1997. Face recognition: a convolutional neural-network approach, IEEE Transactions on Neural Networks, (8-11):98-113.

LeCun, Y., and Bengio, Y., 1995. Convolutional networks for images, speech, and time series (Michael A. Arbib, editor), The Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, Massachusetts, 255–258.

LeCun, Y., Bengio, Y., and Hinton, G. E., 2015. Deep Learning, Nature, 521:436–444

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D., 1990. Handwritten digit recognition with a back-propagation network, In Proc. Advances in Neural Information Processing Systems: 396–404

Liu, H. and Setiono, R., 1995. Chi2: Feature selection and discretization of numeric attributes, IEEE 7th International Conference on Tools with Artificial Intelligence, 388–391.

Liu, H., Motoda, H., 2007. Computational Methods of Feature Selection, Chapman and Hall/CRC.

Mahmood, A., Bennamoun, M., An, S., and Sohel, F., 2016. ResFeats: Residual Network Based Features forImage Classification, arXiv preprint, arXiv:1611.06656

Narendra, P.M., and Fukunaga, K., 1977. A branch and bound algorithm for feature selection, IEEE Transactions on Computers, C-26(9):917–922.

Paul, R., Hawkins, S. H., Balagurunathan, Y., Schabath, M. B., Gillies, R. J., Hall, L. O., and Goldgof, D. B., 2016. Deep Feature Transfer Learning in Combination with Traditional Features Predicts Survival Among Patients with Lung Adenocarcinoma, Tomography : A Journal for Imaging Research, 2(4): 388–395.

Provost, F., 2000. Distributed data mining: Scaling up and beyond, Advances in Distributed and Parallel Knowledge Discovery.

Qian, R., Yue, Y., Coenen, F., and Zhang, B., 2016. Visual attribute classification using feature selection and convolutional neural network, IEEE 13th International Conference on Signal Processing (ICSP), Chengdu, 649-653

Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S., 2014. CNN Features off-the-shelf: an Astounding Baseline for Recognition, arXiv preprint, arXiv:1403.6382v3.

Recker, S., and Gribble, C., 2017. Real-Time, In Situ Intelligent Video Analytics: Harnessing The Power of GPUs for Deep Learning Applications, DSIAC Journals / Winter: Volume 4 Number 1.

Rui, T., Zou, J., Zhou, Y., Fei, J., and Yang, C., 2016. Convolutional Neural Network Simplification Based on Feature Maps Selection, IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), Wuhan, 1207-1210.

Scherer D., Müller A., and Behnke S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition, International conference on artificial neural networks, (6354):92–101.

Simonyan, K., and Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint, arXiv:1409.1556.

Sohn, K., Jung, D., Lee, H., and Hero III, A., 2011. Efficient learning of sparse, distributed, convolutional feature representations for object recognition, ICCV.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting, Journal of machine learning research 15(1): 1929 - 1958.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., 2014. Going deeper with convolutions, arXiv preprint, arXiv:1409.4842.

Wang, H. and Raj, B., 2017. On the Origin of Deep Learning, arXiv preprint, arXiv:1702.07800v4.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Y. Gong, 2010. Locality-constrained linear coding for image classification, CVPR.

Wang, L., Ouyang, W., Wang, X., and Lu, H., 2015. Visual Tracking with Fully Convolutional Networks, IEEE International Conference on Computer Vision (ICCV), Santiago, 3119-3127.

Yang, B., Yan, J., Lei, Z., Li, S. Z., 2015. Convolutional Channel Features, arXiv preprint, arXiv:1504.07339.

Zeiler M. D., and Fergus R., 2013. Stochastic pooling for regularization of deep convolutional neural networks, arXiv preprint, arXiv:1301.3557

Zeiler M. D., and Fergus R., 2014. Visualizing and Understanding Convolutional Networks, arXiv preprint, arXiv:1311.2901.

Zheng, L., Zhao, Y., Wang, S., Wang, J., and Tian, Q., 2016. Good Practice in CNN Feature Transfer, arXiv preprint, arXiv:1604.00133

BIOGRAPHY

Serkan ÖZTÜRK was born on July 22th, 1980 in Adana. He was completed his elementary education at Yavuzlar Primary School and secondary education at Karaisalı health vocational high School. He was graduated from Mersin University in 2007 with a B. S. in Mathematics and from Çukurova University in 2018 with a B. S. in Computer Engineering. He worked as a nurse at 112 Emergency Services, in Adana. He is married to Fatma and the father of Ali Erkin and Duru.

APPENDIX

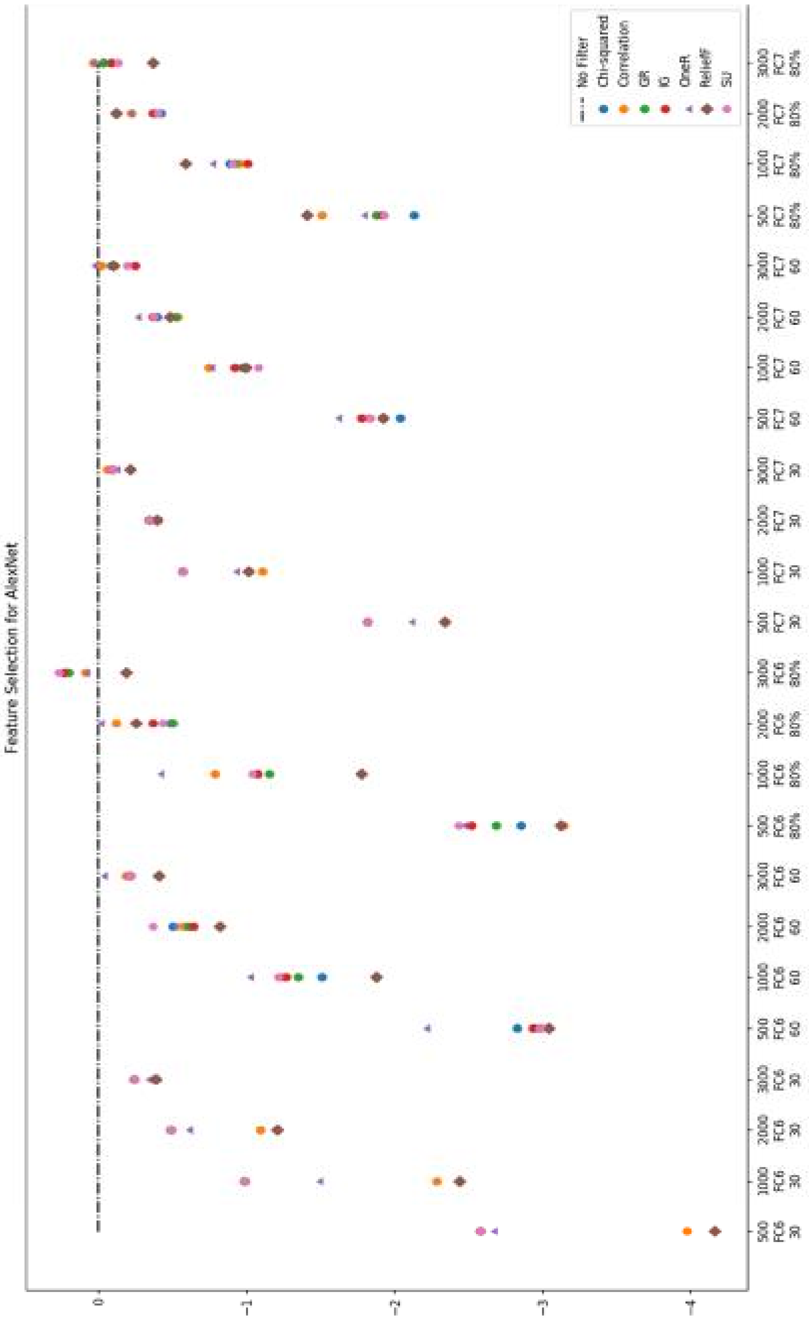
In Appendix A, the results for all the filter selection methods for AlexNet, VGG16, and VGG19 architectures, respectively, are shown in a single image. Thus, it has been shown more clearly how these feature selection methods tend to relate to the training set size and the FC layer in the respective architectures.

In Appendix B, architectural best-worst predictions were made for 15 class F-measure values, which were estimated using all the features.

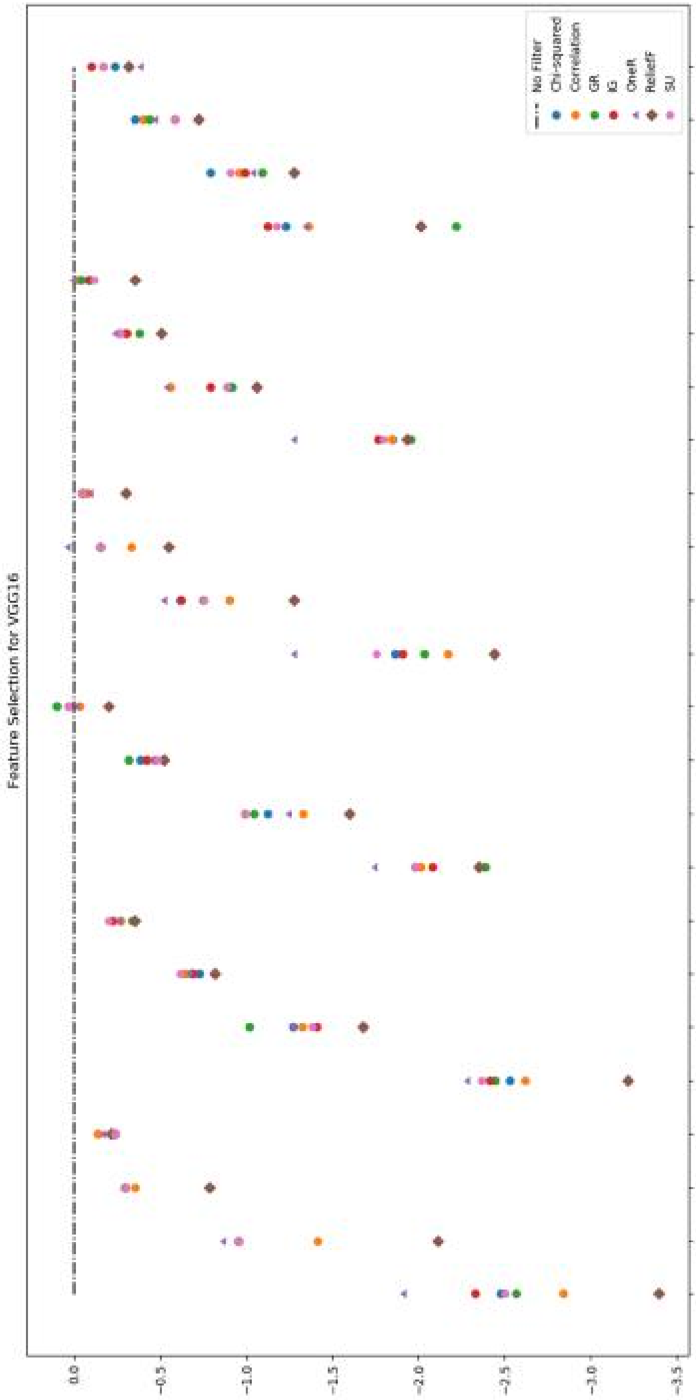
In Appendix C, all of the precision, recall, f-measure and accuracy values of the selected 500, 1000, 2000 and 3000 features are shown.

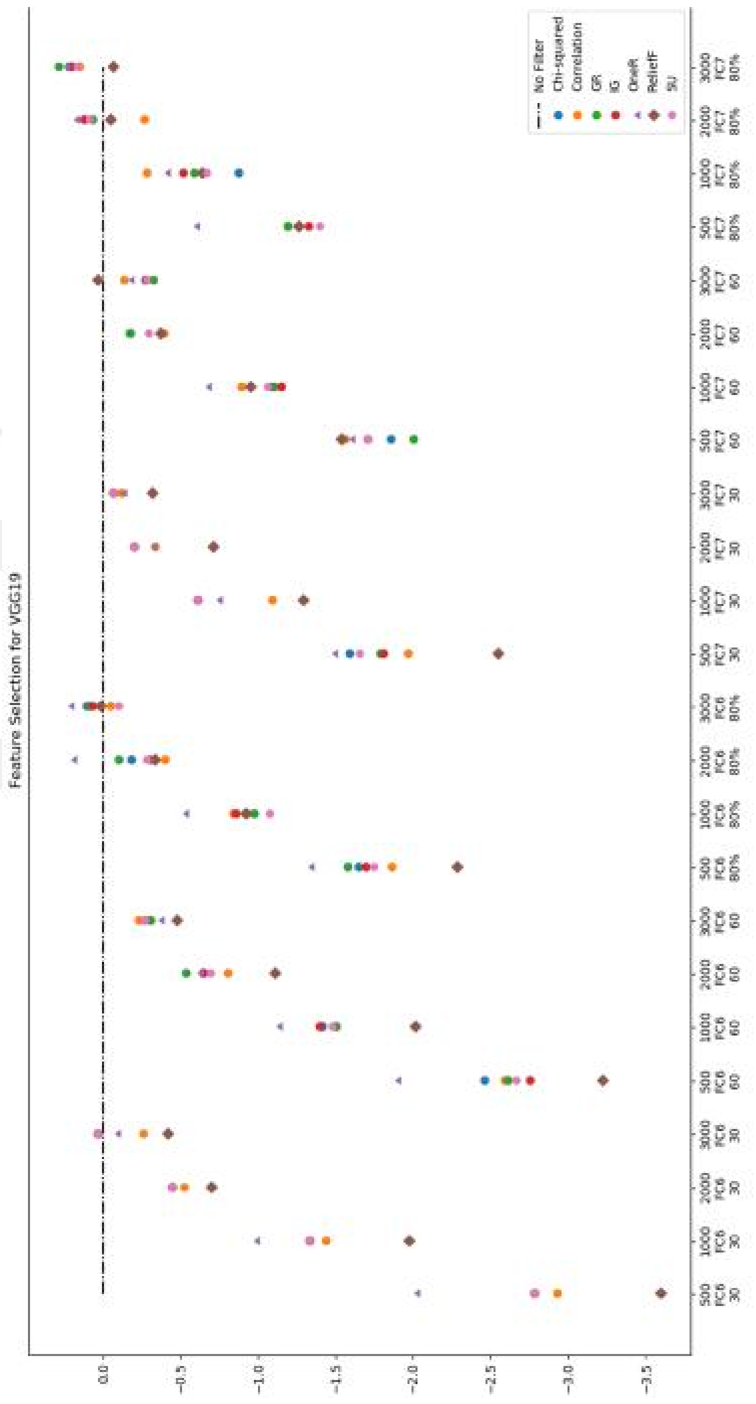
In Appendix D, all of the classes of Caltech-256 dataset are shown.

APPENDIX A



Feature Selection for AlexNet
No Filter
Chi-squared
Correlation
GR
IG
OneR
ReliefF
SU
0
−1
−2
−3
−4


Feature Selection for VGG16
No Filter
Chi-squared
Correlation
GR
IG
OneR
ReliefF
SU
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
500 FC6 30
1000 FC6 30
2000 FC6 30
3000 FC6 30
500 FC6 60
1000 FC6 60
2000 FC6 60
3000 FC6 60
500 FC6 80%
1000 FC6 80%
2000 FC6 80%
3000 FC6 80%
500 FC7 30
1000 FC7 30
2000 FC7 30
3000 FC7 30
500 FC7 60
1000 FC7 60
2000 FC7 60
3000 FC7 60
500 FC7 80%
1000 FC7 80%
2000 FC7 80%
3000 FC7 80%


Feature Selection for VGG19
No Filter
Chi-squared
Correlation
GR
IG
OneR
ReliefF
SU

APPENDIX B

The F-measure value is the smallest 15 classes for AlexNet FC6 with 30 per class			
Precision	Recall	F-Measure	Class
0.1	0.145	0.118	rifle
0.162	0.153	0.157	screwdriver
0.155	0.17	0.162	drinking-straw
0.155	0.178	0.166	skateboard
0.182	0.211	0.195	soda-can
0.219	0.229	0.224	tuning-fork
0.173	0.367	0.235	playing-card
0.269	0.222	0.243	mailbox
0.206	0.296	0.243	knife
0.294	0.208	0.244	dumb-bell
0.227	0.309	0.262	hot-dog
0.256	0.292	0.273	sword
0.245	0.311	0.274	canoe
0.256	0.3	0.276	yo-yo
0.255	0.316	0.282	bat

The F-measure value is the largest 15 classes for AlexNet FC6 with 30 per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
0.992	0.951	0.971	faces
0.996	0.892	0.941	airplanes
0.928	0.951	0.939	ketch
0.994	0.882	0.934	motorbikes
0.959	0.888	0.922	leopards
0.885	0.92	0.902	sunflower
0.919	0.864	0.891	zebra
0.882	0.889	0.885	mars
0.857	0.913	0.884	bonsai
0.811	0.938	0.87	trilobite
0.971	0.772	0.86	watch
0.891	0.831	0.86	menorah
0.809	0.917	0.859	tower-pisa
0.787	0.944	0.859	self-propelled-lawn-mower

The F-measure value is the smallest 15 classes for VGG16 FC6 with 30 per class			
Precision	Recall	F-Measure	Class
0.25	0.2	0.222	chopsticks
0.257	0.271	0.264	tuning-fork
0.178	0.536	0.267	fighter-jet
0.233	0.316	0.268	rifle
0.238	0.358	0.286	drinking-straw
0.244	0.517	0.332	playing-card
0.447	0.278	0.343	ladder
0.364	0.329	0.345	skateboard
0.368	0.357	0.362	yo-yo
0.313	0.481	0.379	minotaur
0.417	0.347	0.379	screwdriver
0.337	0.437	0.38	knife
0.341	0.431	0.38	sword
0.342	0.439	0.385	jesus-christ
0.393	0.381	0.387	mailbox

The F-measure value is the largest 15 classes for VGG16 FC6 with 30 per class			
Precision	Recall	F-Measur	Class
0.989	1.000	0.994	car-side
0.975	0.963	0.969	ketch
1.000	0.919	0.958	french-horn
0.938	0.953	0.946	trilobite
0.923	0.96	0.941	sunflower
1.000	0.862	0.926	motorbikes
0.972	0.875	0.921	leopards
0.986	0.859	0.918	faces
0.93	0.898	0.914	hot-air-balloon
0.951	0.879	0.913	zebra
0.891	0.932	0.911	fire-truck
0.923	0.882	0.902	school-bus
0.976	0.823	0.893	billiards
0.854	0.929	0.89	mars
0.979	0.813	0.888	watch

The F-measure value is the smallest 15 classes for VGG19 FC6 with 30 per class			
Precision	Recall	F-Measure	Class
0.238	0.283	0.259	drinking-straw
0.197	0.383	0.26	playing-card
0.227	0.355	0.277	rifle
0.193	0.594	0.292	fighter-jet
0.366	0.273	0.312	chopsticks
0.471	0.269	0.342	ladder
0.314	0.386	0.346	tuning-fork
0.28	0.491	0.356	floppy-disk
0.329	0.404	0.362	soda-can
0.35	0.382	0.365	hot-dog
0.329	0.474	0.388	jesus-christ
0.419	0.361	0.388	screwdriver
0.378	0.4	0.389	yo-yo
0.29	0.623	0.396	picnic-table
0.429	0.381	0.403	mailbox

The F-measure value is the largest 15 classes for VGG19 FC6 with 30 per class			
Precision	Recall	F-Measure	Class
0.966	1.000	0.983	car-side
0.983	0.952	0.967	french-horn
0.963	0.963	0.963	ketch
0.952	0.938	0.945	trilobite
0.984	0.909	0.945	zebra
0.999	0.876	0.933	motorbikes
0.94	0.922	0.931	skunk
0.986	0.872	0.925	faces
0.946	0.898	0.922	hot-air-balloon
0.935	0.894	0.914	leopards
0.986	0.848	0.912	watch
0.899	0.909	0.904	fire-truck
0.918	0.882	0.9	starfish
0.86	0.929	0.893	mars
0.953	0.823	0.883	billiards

The F-measure value is the smallest 15 classes for AlexNet FC7 with 30 per class			
Precision	Recall	F-Measure	Class
0.156	0.184	0.169	rifle
0.212	0.175	0.191	mailbox
0.267	0.171	0.209	tuning-fork
0.262	0.2	0.227	chopsticks
0.185	0.338	0.239	knife
0.189	0.333	0.241	playing-card
0.213	0.286	0.244	yo-yo
0.266	0.236	0.25	sword
0.223	0.288	0.251	skateboard
0.227	0.283	0.252	drinking-straw
0.28	0.255	0.267	hot-dog
0.298	0.246	0.269	soda-can
0.259	0.284	0.271	canoe
0.3	0.25	0.273	screwdriver
0.442	0.217	0.291	ladder

The F-measure value is the largest 15 classes for AlexNet FC7 with 30 per class			
Precision	Recall	F-Measure	Class
0.966	1.000	0.983	car-side
0.963	0.963	0.963	ketch
0.987	0.868	0.923	airplanes
0.911	0.932	0.921	fire-truck
0.991	0.849	0.915	faces
0.873	0.96	0.914	sunflower
0.951	0.879	0.913	zebra
0.992	0.806	0.89	leopards
0.851	0.905	0.877	mars
0.992	0.772	0.868	motorbikes
0.862	0.875	0.868	trilobite
0.836	0.897	0.865	school-bus
0.804	0.935	0.864	bonsai
0.789	0.956	0.864	self-propelled-lawn-mower
0.938	0.795	0.861	watch

The F-measure value is the smallest 15 classes for VGG16 FC7 with 30 per class			
Precision	Recall	F-Measure	Class
0.194	0.245	0.217	drinking-straw
0.245	0.218	0.231	chopsticks
0.25	0.3	0.273	tuning-fork
0.427	0.208	0.279	ladder
0.232	0.365	0.284	mailbox
0.25	0.367	0.297	playing-card
0.211	0.623	0.315	fighter-jet
0.299	0.342	0.319	rifle
0.316	0.343	0.329	yo-yo
0.286	0.479	0.358	knife
0.333	0.4	0.364	hot-dog
0.377	0.39	0.383	harmonica
0.419	0.361	0.388	screwdriver
0.313	0.544	0.397	coffin
0.319	0.526	0.397	jesus-christ

The F-measure value is the largest 15 classes for VGG16 FC7 with 30 per class			
Precision	Recall	F-Measure	Class
0.966	0.836	0.897	watch
0.923	0.882	0.902	school-bus
0.846	0.978	0.907	self-propelled-lawn-mower
0.983	0.847	0.91	faces
0.889	0.941	0.914	skunk
0.997	0.845	0.915	motorbikes
0.979	0.863	0.917	leopards
0.889	0.96	0.923	sunflower
0.897	0.953	0.924	trilobite
0.932	0.932	0.932	hot-air-balloon
0.922	0.943	0.933	fire-truck
0.968	0.909	0.937	zebra
0.983	0.935	0.959	french-horn
0.975	0.951	0.963	ketch
0.966	1.000	0.983	car-side

The F-measure value is the smallest 15 classes for VGG19 FC7 with 30 per class			
Precision	Recall	F-Measure	Class
0.283	0.236	0.257	chopsticks
0.258	0.321	0.286	drinking-straw
0.266	0.309	0.286	hot-dog
0.383	0.257	0.308	tuning-fork
0.505	0.231	0.317	ladder
0.252	0.483	0.331	playing-card
0.226	0.652	0.336	fighter-jet
0.291	0.421	0.344	rifle
0.333	0.357	0.345	yo-yo
0.29	0.474	0.36	jesus-christ
0.293	0.469	0.361	people
0.4	0.356	0.377	skateboard
0.347	0.456	0.394	soda-can
0.4	0.413	0.406	mailbox
0.319	0.558	0.406	minotaur

The F-measure value is the largest 15 classes for VGG19 FC7 with 30 per class			
Precision	Recall	F-Measure	Class
0.966	0.988	0.977	car-side
0.983	0.952	0.967	french-horn
0.954	0.969	0.961	trilobite
0.951	0.963	0.957	ketch
0.984	0.909	0.945	zebra
0.932	0.932	0.932	fire-truck
0.94	0.922	0.931	skunk
0.995	0.863	0.925	motorbikes
0.941	0.9	0.92	leopards
0.881	0.937	0.908	hawksbill
0.929	0.881	0.904	hot-air-balloon
0.923	0.882	0.902	school-bus
0.873	0.923	0.897	kangaroo
0.885	0.902	0.893	starfish
0.947	0.842	0.892	watch

The F-measure value is the smallest 15 classes for AlexNet FC6 with 60 per class			
Precision	Recall	F-Measure	Class
0.077	0.087	0.082	drinking-straw
0.119	0.174	0.142	rifle
0.16	0.148	0.154	soda-can
0.171	0.28	0.212	hot-dog
0.225	0.225	0.225	tuning-fork
0.267	0.242	0.254	mailbox
0.265	0.29	0.277	wheelbarrow
0.23	0.35	0.277	yo-yo
0.333	0.238	0.278	screwdriver
0.277	0.289	0.283	spoon
0.308	0.279	0.293	skateboard
0.224	0.433	0.295	playing-card
0.254	0.364	0.299	canoe
0.333	0.279	0.304	conch
0.22	0.542	0.313	refrigerator

The F-measure value is the largest 15 classes for AlexNet FC6 with 60 per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
0.992	1.000	0.996	leopards
0.994	0.96	0.977	faces
0.996	0.942	0.968	motorbikes
0.997	0.893	0.942	airplanes
0.894	0.983	0.937	self-propelled-lawn-mower
0.919	0.944	0.932	zebra
0.87	1.000	0.93	sunflower
0.906	0.941	0.923	ketch
0.875	0.948	0.91	mars
0.857	0.968	0.909	bonsai
0.829	1.000	0.907	trilobite
0.857	0.931	0.893	fire-truck
0.926	0.862	0.893	menorah
0.844	0.931	0.885	hot-air-balloon

The F-measure value is the smallest 15 classes for VGG16 FC6 with 60 per class			
Precision	Recall	F-Measure	Class
0.122	0.615	0.203	fighter-jet
0.211	0.348	0.262	drinking-straw
0.239	0.5	0.324	mountain-bike
0.289	0.367	0.324	playing-card
0.303	0.4	0.345	hot-dog
0.313	0.4	0.351	chopsticks
0.321	0.391	0.353	rifle
0.289	0.481	0.361	jesus-christ
0.286	0.593	0.386	coffin
0.367	0.407	0.386	soda-can
0.281	0.643	0.391	smokestack
0.429	0.375	0.4	yo-yo
0.375	0.439	0.404	knife
0.604	0.319	0.417	ladder
0.417	0.417	0.417	theodolite

The F-measure value is the largest 15 classes for VGG16 FC6 with 60 per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
1.000	1.000	1.000	trilobite
0.956	0.992	0.974	leopards
0.98	0.961	0.97	ketch
0.966	0.966	0.966	hot-air-balloon
0.989	0.941	0.964	faces
0.971	0.944	0.958	zebra
0.957	0.957	0.957	eiffel-tower
1.000	0.906	0.951	french-horn
0.921	0.969	0.944	mars
0.998	0.881	0.936	motorbikes
0.966	0.903	0.933	license-plate
0.971	0.895	0.932	school-bus
0.931	0.931	0.931	menorah
0.95	0.905	0.927	skunk

The F-measure value is the smallest 15 classes for VGG19 FC6 with 60 per class			
Precision	Recall	F-Measure	Class
0.172	0.5	0.256	mountain-bike
0.17	0.769	0.279	fighter-jet
0.281	0.391	0.327	drinking-straw
0.313	0.37	0.339	soda-can
0.303	0.4	0.345	chopsticks
0.262	0.533	0.352	playing-card
0.3	0.45	0.36	yo-yo
0.295	0.481	0.366	jesus-christ
0.309	0.457	0.368	rifle
0.389	0.438	0.412	paperclip
0.396	0.432	0.413	canoe
0.575	0.335	0.424	ladder
0.342	0.565	0.426	floppy-disk
0.396	0.5	0.442	screwdriver
0.476	0.417	0.444	theodolite

The F-measure value is the largest 15 classes for VGG19 FC6 with 60 per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
0.971	1.000	0.986	trilobite
0.956	1.000	0.977	leopards
0.989	0.939	0.963	faces
0.971	0.944	0.958	zebra
0.927	0.974	0.95	revolver
0.93	0.969	0.949	mars
0.915	0.977	0.945	guitar-pick
0.998	0.888	0.94	motorbikes
1.000	0.875	0.933	french-horn
0.946	0.921	0.933	school-bus
0.94	0.922	0.931	ketch
0.95	0.905	0.927	starfish
0.968	0.882	0.923	laptop
0.992	0.851	0.916	watch

The F-measure value is the smallest 15 classes for AlexNet FC7 with 60 per class			
Precision	Recall	F-Measure	Class
0.073	0.13	0.094	drinking-straw
0.185	0.185	0.185	soda-can
0.191	0.273	0.225	mailbox
0.188	0.295	0.23	canoe
0.207	0.261	0.231	rifle
0.211	0.267	0.235	playing-card
0.231	0.3	0.261	yo-yo
0.209	0.36	0.265	chopsticks
0.357	0.222	0.274	spoon
0.225	0.36	0.277	hot-dog
0.324	0.262	0.289	dumb-bell
0.353	0.269	0.305	baseball-bat
0.325	0.302	0.313	skateboard
0.25	0.419	0.313	wheelbarrow
0.286	0.357	0.317	pram

The F-measure value is the largest 15 classes for AlexNet FC7 with 60 per class			
Precision	Recall	F-Measure	Class
0.982	1.000	0.991	car-side
0.984	0.969	0.977	leopards
0.952	0.983	0.967	self-propelled-lawn-mower
0.98	0.941	0.96	ketch
0.994	0.92	0.956	motorbikes
0.988	0.912	0.949	faces
0.943	0.917	0.93	zebra
0.992	0.862	0.923	airplanes
0.868	0.971	0.917	trilobite
0.881	0.952	0.915	bonsai
0.9	0.931	0.915	fire-truck
0.899	0.927	0.913	mars
0.833	1.000	0.909	sunflower
0.839	0.897	0.867	menorah
0.792	0.95	0.864	golden-gate-bridge

The F-measure value is the smallest 15 classes for VGG16 FC7 with 60 per class			
Precision	Recall	F-Measure	Class
0.172	0.217	0.192	drinking-straw
0.148	0.667	0.242	fighter-jet
0.258	0.32	0.286	chopsticks
0.327	0.391	0.356	rifle
0.298	0.467	0.364	playing-card
0.514	0.297	0.376	ladder
0.333	0.45	0.383	yo-yo
0.314	0.5	0.386	mountain-bike
0.311	0.519	0.389	jesus-christ
0.35	0.438	0.389	paperclip
0.375	0.409	0.391	canoe
0.353	0.444	0.393	soda-can
0.324	0.545	0.407	minotaur
0.5	0.349	0.411	skateboard
0.375	0.48	0.421	hot-dog

The F-measure value is the largest 15 classes for VGG16 FC7 with 60 per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	trilobite
0.982	1.000	0.991	car-side
0.962	0.985	0.973	leopards
0.975	0.947	0.961	faces
0.98	0.941	0.96	ketch
0.946	0.972	0.959	zebra
0.909	1.000	0.952	sunflower
0.933	0.966	0.949	hot-air-balloon
0.999	0.904	0.949	motorbikes
0.896	1.000	0.945	self-propelled-lawn-mower
0.893	1.000	0.943	hourglass
0.891	0.983	0.934	fire-truck
1.000	0.875	0.933	french-horn
0.995	0.867	0.926	billiards
0.972	0.875	0.921	harp

The F-measure value is the smallest 15 classes for VGG19 FC7 with 60 per class			
Precision	Recall	F-Measure	Class
0.137	0.718	0.23	fighter-jet
0.226	0.304	0.259	drinking-straw
0.28	0.28	0.28	chopsticks
0.265	0.409	0.321	mountain-bike
0.275	0.407	0.328	jesus-christ
0.405	0.341	0.37	canoe
0.294	0.5	0.37	playing-card
0.333	0.444	0.381	soda-can
0.308	0.5	0.381	yo-yo
0.549	0.308	0.394	ladder
0.338	0.5	0.404	rifle
0.417	0.417	0.417	theodolite
0.377	0.483	0.424	people
0.357	0.536	0.429	pram
0.341	0.583	0.431	refrigerator

The F-measure value is the largest 15 classes for VGG19 FC7 with 60 per class			
Precision	Recall	F-Measure	Class
0.982	1.000	0.991	car-side
0.955	0.985	0.97	leopards
0.943	0.971	0.957	trilobite
0.991	0.923	0.956	faces
0.999	0.909	0.952	motorbikes
0.942	0.961	0.951	ketch
0.925	0.974	0.949	school-bus
0.921	0.972	0.946	zebra
0.904	0.979	0.94	mars
0.903	0.966	0.933	fire-truck
1.000	0.875	0.933	french-horn
0.902	0.939	0.92	swiss-army-knife
0.886	0.939	0.912	hawksbill
0.962	0.864	0.911	llama
0.921	0.897	0.909	revolver

The F-measure value is the smallest 15 classes for AlexNet FC6 with 80% of per class			
Precision	Recall	F-Measure	Class
0.154	0.118	0.133	drinking-straw
0.4	0.105	0.167	mailbox
0.19	0.19	0.19	rifle
0.222	0.19	0.205	canoe
0.333	0.15	0.207	dumb-bell
0.333	0.15	0.207	screwdriver
0.25	0.273	0.261	goat
0.308	0.235	0.267	soda-can
0.4	0.235	0.296	hot-dog
0.333	0.286	0.308	spoon
0.353	0.286	0.316	bat
0.316	0.333	0.324	paperclip
0.304	0.35	0.326	sword
0.333	0.353	0.343	jesus-christ
0.286	0.444	0.348	playing-card

The F-measure value is the largest 15 classes for AlexNet FC6 with 80% of per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
1.000	1.000	1.000	leopards
0.988	0.994	0.991	airplanes
0.982	1.000	0.991	motorbikes
0.96	1.000	0.98	fire-truck
0.952	1.000	0.976	school-bus
1.000	0.944	0.971	tower-pisa
0.941	1.000	0.97	sunflower
0.923	1.000	0.96	self-propelled-lawn-mower
0.958	0.958	0.958	bonsai
0.944	0.966	0.955	faces
0.88	1.000	0.936	ketch
0.952	0.909	0.93	hibiscus
0.941	0.889	0.914	menorah
0.833	1.000	0.909	porcupine

The F-measure value is the smallest 15 classes for VGG16 FC6 with 80% of per class			
Precision	Recall	F-Measure	Class
0.308	0.235	0.267	drinking-straw
0.455	0.278	0.345	paperclip
0.368	0.333	0.35	rifle
0.364	0.444	0.4	playing-card
0.444	0.381	0.41	skateboard
0.5	0.389	0.438	wheelbarrow
0.375	0.529	0.439	floppy-disk
0.455	0.435	0.444	frog
0.6	0.353	0.444	hot-dog
0.5	0.412	0.452	soda-can
0.393	0.55	0.458	yo-yo
0.471	0.471	0.471	coffin
0.382	0.619	0.473	people
0.429	0.545	0.48	sneaker
0.7	0.368	0.483	mailbox

The F-measure value is the largest 15 classes for VGG16 FC6 with 80% of per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
1.000	1.000	1.000	eiffel-tower
1.000	1.000	1.000	hourglass
1.000	1.000	1.000	sunflower
1.000	1.000	1.000	trilobite
0.976	1.000	0.988	motorbikes
0.958	0.994	0.975	airplanes
0.95	1.000	0.974	leopards
1.000	0.95	0.974	school-bus
0.941	1.000	0.97	desk-globe
1.000	0.941	0.97	ewer
0.929	1.000	0.963	elephant
0.963	0.963	0.963	teapot
0.955	0.966	0.96	faces
0.923	1.000	0.96	fire-truck

The F-measure value is the smallest 15 classes for VGG19 FC6 with 80% of per class			
Precision	Recall	F-Measure	Class
0.429	0.333	0.375	paperclip
0.667	0.286	0.4	canoe
0.462	0.353	0.4	drinking-straw
0.462	0.353	0.4	soda-can
0.625	0.313	0.417	mountain-bike
0.444	0.4	0.421	knife
0.438	0.412	0.424	refrigerator
0.467	0.389	0.424	wheelbarrow
0.467	0.389	0.424	windmill
0.545	0.353	0.429	theodolite
0.45	0.429	0.439	rifle
0.5	0.409	0.45	sneaker
0.571	0.381	0.457	skateboard
0.4	0.556	0.465	playing-card
0.471	0.471	0.471	coffin

The F-measure value is the largest 15 classes for VGG19 FC6 with 80% of per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
1.000	1.000	1.000	revolver
1.000	1.000	1.000	sunflower
0.976	1.000	0.988	motorbikes
0.96	1.000	0.98	fire-truck
0.969	0.988	0.978	airplanes
1.000	0.95	0.974	school-bus
0.95	1.000	0.974	trilobite
0.944	1.000	0.971	hourglass
1.000	0.944	0.971	tower-pisa
0.966	0.966	0.966	faces
0.927	1.000	0.962	leopards
1.000	0.923	0.96	elephant
0.923	1.000	0.96	self-propelled-lawn-mower
0.917	1.000	0.957	ketch

The F-measure value is the smallest 15 classes for AlexNet FC7 with 80% of per class			
Precision	Recall	F-Measure	Class
0.167	0.059	0.087	drinking-straw
0.182	0.095	0.125	rifle
0.231	0.176	0.2	floppy-disk
0.25	0.182	0.211	goat
0.375	0.176	0.24	soda-can
0.24	0.333	0.279	playing-card
0.235	0.421	0.302	mailbox
0.385	0.25	0.303	screwdriver
0.353	0.286	0.316	canoe
0.333	0.3	0.316	knife
0.333	0.353	0.343	hot-dog
0.333	0.412	0.368	jesus-christ
0.458	0.314	0.373	mussels
0.321	0.45	0.375	yo-yo
0.438	0.333	0.378	spoon

The F-measure value is the largest 15 classes for AlexNet FC7 with 80% of per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
1.000	1.000	1.000	leopards
0.976	1.000	0.988	motorbikes
0.969	0.988	0.978	airplanes
0.957	1.000	0.978	ketch
0.952	1.000	0.976	school-bus
0.95	1.000	0.974	hawksbill
0.977	0.966	0.971	faces
0.941	1.000	0.97	sunflower
0.958	0.958	0.958	bonsai
1.000	0.917	0.957	fire-truck
0.905	1.000	0.95	trilobite
1.000	0.889	0.941	tower-pisa
0.92	0.958	0.939	self-propelled-lawn-mower
0.944	0.895	0.919	zebra

The F-measure value is the smallest 15 classes for VGG16 FC7 with 80% of per class			
Precision	Recall	F-Measure	Class
0.389	0.333	0.359	rifle
0.375	0.353	0.364	drinking-straw
0.462	0.333	0.387	paperclip
0.391	0.45	0.419	screwdriver
0.471	0.381	0.421	canoe
0.438	0.412	0.424	coffin
0.438	0.412	0.424	soda-can
0.455	0.4	0.426	baseball-bat
0.5	0.381	0.432	skateboard
0.533	0.4	0.457	sword
0.75	0.333	0.462	pram
0.415	0.524	0.463	people
0.533	0.421	0.471	mailbox
0.615	0.4	0.485	dice
0.6	0.429	0.5	bat

The F-measure value is the largest 15 classes for VGG16 FC7 with 80% of per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
0.974	1.000	0.987	leopards
0.97	1.000	0.985	motorbikes
0.957	1.000	0.978	ketch
0.952	1.000	0.976	school-bus
0.95	1.000	0.974	trilobite
0.944	1.000	0.971	hourglass
0.941	1.000	0.97	sunflower
0.935	0.988	0.96	airplanes
0.958	0.958	0.958	fire-truck
0.938	0.968	0.952	hot-tub
0.933	0.966	0.949	faces
0.903	1.000	0.949	teepee
0.917	0.982	0.948	billiards
0.944	0.944	0.944	hot-air-balloon

The F-measure value is the smallest 15 classes for VGG19 FC7 with 80% of per class			
Precision	Recall	F-Measure	Class
0.2	0.176	0.188	drinking-straw
0.5	0.188	0.273	mountain-bike
0.455	0.238	0.312	canoe
0.353	0.353	0.353	coffin
0.5	0.278	0.357	paperclip
0.333	0.389	0.359	playing-card
0.5	0.294	0.37	theodolite
0.5	0.316	0.387	mailbox
0.75	0.3	0.429	dice
0.5	0.381	0.432	cake
0.455	0.435	0.444	frog
0.538	0.412	0.467	refrigerator
0.7	0.35	0.467	sushi
0.411	0.548	0.469	people
0.5	0.45	0.474	screwdriver

The F-measure value is the largest 15 classes for VGG19 FC7 with 80% of per class			
Precision	Recall	F-Measure	Class
1.000	1.000	1.000	car-side
1.000	1.000	1.000	fire-truck
1.000	1.000	1.000	school-bus
1.000	1.000	1.000	sunflower
0.97	1.000	0.985	motorbikes
0.981	0.988	0.984	airplanes
0.957	1.000	0.978	ketch
0.95	1.000	0.974	trilobite
1.000	0.947	0.973	hawksbill
1.000	0.941	0.97	rotary-phone
0.976	0.954	0.965	faces
0.927	1.000	0.962	leopards
1.000	0.9	0.947	revolver
0.947	0.947	0.947	saturn
0.895	1.000	0.944	hourglass

APPENDIX C

AlexNet FC6 with 30 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	58.543	60.1357	60.6335	60.8824
	Precision	61.7	63.1	63.6	63.8
	Recall	58.5	60.1	60.6	60.9
	F-measure	59.1	60.6	61.1	61.4
Correlation Ranking Filter	Accuracy	57.1448	58.8371	60.0226	60.7421
	Precision	60.5	62.0	63.0	63.6
	Recall	57.1	58.8	60.0	60.7
	F-measure	57.8	59.4	60.5	61.2
Gain Ratio feature evaluator	Accuracy	58.543	60.1357	60.6335	60.8824
	Precision	61.7	63.1	63.6	63.8
	Recall	58.5	60.1	60.6	60.9
	F-measure	59.1	60.6	61.1	61.4
Information Gain Ranking Filter	Accuracy	58.543	60.1357	60.6335	60.8824
	Precision	61.7	63.1	63.6	63.8
	Recall	58.5	60.1	60.6	60.9
	F-measure	59.1	60.6	61.1	61.4
OneR feature evaluator	Accuracy	58.4661	59.6425	60.5249	60.8009
	Precision	61.4	62.6	63.3	63.6
	Recall	58.5	59.6	60.5	60.8
	F-measure	58.9	60.1	61.0	61.3
ReliefF Ranking Filter	Accuracy	56.9593	58.6787	59.9095	60.7376
	Precision	60.3	61.8	62.8	63.7
	Recall	57.0	58.7	59.9	60.7
	F-measure	57.6	59.3	60.4	61.2
Symmetrical Uncertainty Ranking Filter	Accuracy	58.543	60.1357	60.6335	60.8824
	Precision	61.7	63.1	63.6	63.8
	Recall	58.5	60.1	60.6	60.9
	F-measure	59.1	60.6	61.1	61.4
No Filter	Accuracy	61.1222			
	Precision	64.0			
	Recall	61.1			
	F-measure	61.6			

AlexNet FC6 with 60 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	63.7101	65.0277	66.0402	66.3245
	Precision	67.5	68.5	69.5	69.6
	Recall	63.7	65.0	66.0	66.3
	F-measure	64.4	65.5	66.6	66.8
Correlation Ranking Filter	Accuracy	63.5506	65.3121	65.9639	66.3523
	Precision	67.4	68.7	69.5	69.6
	Recall	63.6	65.3	66.0	66.4
	F-measure	64.2	65.8	66.5	66.8
Gain Ratio feature evaluator	Accuracy	63.5645	65.1872	65.9293	66.3245
	Precision	67.2	68.7	69.4	69.6
	Recall	63.6	65.2	65.9	66.3
	F-measure	64.2	65.7	66.5	66.8
Information Gain Ranking Filter	Accuracy	63.6061	65.2705	65.8946	66.3245
	Precision	67.3	68.8	69.3	69.6
	Recall	63.6	65.3	65.9	66.3
	F-measure	64.3	65.8	66.5	66.8
OneR feature evaluator	Accuracy	64.3412	65.527	66.0402	66.5187
	Precision	67.8	68.8	69.4	69.8
	Recall	64.3	65.5	66.0	66.5
	F-measure	64.9	65.9	66.5	67.0
ReliefF Ranking Filter	Accuracy	63.4951	64.6602	65.7143	66.1304
	Precision	67.2	68.3	69.2	69.4
	Recall	63.5	64.7	65.7	66.1
	F-measure	64.1	65.2	66.3	66.7
Symmetrical Uncertainty Ranking Filter	Accuracy	63.5576	65.3259	66.172	66.3245
	Precision	67.3	68.8	69.6	69.6
	Recall	63.6	65.3	66.2	66.3
	F-measure	64.2	65.8	66.7	66.8
No Filter	Accuracy	66.5395			
	Precision	69.8			
	Recall	66.5			
	F-measure	67.0			

AlexNet FC6 with 80% of images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	65.5242	67.3051	67.8931	68.5988
	Precision	66.5	68.1	68.8	69.6
	Recall	65.5	67.3	67.9	68.6
	F-measure	65.3	67.0	67.6	68.3
Correlation Ranking Filter	Accuracy	65.2386	67.5907	68.2628	68.4644
	Precision	66.2	68.4	69.1	69.4
	Recall	65.2	67.6	68.3	68.5
	F-measure	64.9	67.2	68.0	68.2
Gain Ratio feature evaluator	Accuracy	65.6922	67.2211	67.8763	68.582
	Precision	66.5	68.0	68.8	69.5
	Recall	65.7	67.2	67.9	68.6
	F-measure	65.3	66.9	67.6	68.3
Information Gain Ranking Filter	Accuracy	65.8602	67.3051	68.0108	68.6156
	Precision	66.8	68.2	68.9	69.6
	Recall	65.9	67.3	68.0	68.6
	F-measure	65.6	67.0	67.7	68.4
OneR feature evaluator	Accuracy	65.9274	67.9772	68.3804	68.4812
	Precision	66.5	68.7	69.0	69.2
	Recall	65.9	68.0	68.4	68.5
	F-measure	65.5	67.6	68.0	68.2
ReliefF Ranking Filter	Accuracy	65.2554	66.5995	68.1284	68.1956
	Precision	66.0	67.3	68.9	69.2
	Recall	65.3	66.6	68.1	68.2
	F-measure	64.9	66.3	67.8	67.9
Symmetrical Uncertainty Ranking Filter	Accuracy	65.9442	67.3387	67.9435	68.6492
	Precision	66.8	68.3	68.8	69.6
	Recall	65.9	67.3	67.9	68.6
	F-measure	65.7	67.0	67.7	68.4
No Filter	Accuracy	68.3804			
	Precision	69.3			
	Recall	68.4			
	F-measure	68.1			

AlexNet FC7 with 30 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	60.6516	61.9005	62.1312	62.3801
	Precision	63.6	64.7	65.0	65.2
	Recall	60.7	61.9	62.1	62.4
	F-measure	61.2	62.4	62.6	62.9
Correlation Ranking Filter	Accuracy	60.1222	61.3620	62.1222	62.4072
	Precision	63.4	64.4	65.1	65.3
	Recall	60.1	61.4	62.1	62.4
	F-measure	60.7	61.9	62.6	62.9
Gain Ratio feature evaluator	Accuracy	60.6516	61.9005	62.1312	62.3801
	Precision	63.6	64.7	65.0	65.2
	Recall	60.7	61.9	62.1	62.4
	F-measure	61.2	62.4	62.6	62.9
Information Gain Ranking Filter	Accuracy	60.6516	61.9005	62.1312	62.3801
	Precision	63.6	64.7	65.0	65.2
	Recall	60.7	61.9	62.1	62.4
	F-measure	61.2	62.4	62.6	62.9
OneR feature evaluator	Accuracy	60.3710	61.5566	62.086	62.3665
	Precision	63.4	64.3	64.9	65.3
	Recall	60.4	61.6	62.1	62.4
	F-measure	60.9	62.1	62.6	62.9
ReliefF Ranking Filter	Accuracy	60.1312	61.4525	62.0769	62.2579
	Precision	63.3	64.5	65.2	65.2
	Recall	60.1	61.5	62.1	62.3
	F-measure	60.7	62.0	62.6	62.8
Symmetrical Uncertainty Ranking Filter	Accuracy	60.6516	61.9005	62.1312	62.3801
	Precision	63.6	64.7	65.0	65.2
	Recall	60.7	61.9	62.1	62.4
	F-measure	61.2	62.4	62.6	62.9
No Filter	Accuracy	62.4706			
	Precision	65.3			
	Recall	62.5			
	F-measure	63.0			

AlexNet FC7 with 60 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	65.3537	66.4147	66.9972	67.3024
	Precision	69.3	70.2	70.8	70.9
	Recall	65.4	66.4	67.0	67.3
	F-measure	66.1	67.2	67.7	68.0
Correlation Ranking Filter	Accuracy	65.5617	66.6436	66.8585	67.3717
	Precision	69.4	70.3	70.6	71.0
	Recall	65.6	66.6	66.9	67.4
	F-measure	66.4	67.4	67.6	68.1
Gain Ratio feature evaluator	Accuracy	65.4716	66.387	66.8724	67.2885
	Precision	69.5	70.3	70.7	70.9
	Recall	65.5	66.4	66.9	67.3
	F-measure	66.3	67.2	67.6	68.0
Information Gain Ranking Filter	Accuracy	65.6172	66.4702	67.025	67.1429
	Precision	69.4	70.3	70.8	70.8
	Recall	65.6	66.5	67.0	67.1
	F-measure	66.3	67.2	67.7	67.8
OneR feature evaluator	Accuracy	65.7906	66.6436	67.1429	67.4411
	Precision	69.6	70.2	70.8	71.2
	Recall	65.8	66.6	67.1	67.4
	F-measure	66.5	67.3	67.9	68.2
ReliefF Ranking Filter	Accuracy	65.4646	66.3939	66.914	67.2954
	Precision	69.4	70.2	70.9	71.0
	Recall	65.5	66.4	66.9	67.3
	F-measure	66.2	67.1	67.7	68.0
Symmetrical Uncertainty Ranking Filter	Accuracy	65.5548	66.3107	67.018	67.1983
	Precision	69.3	70.2	70.8	70.8
	Recall	65.6	66.3	67.0	67.2
	F-measure	66.3	67.1	67.7	67.9
No Filter	Accuracy	67.3925			
	Precision	71.0			
	Recall	67.4			
	F-measure	68.1			

AlexNet FC7 with 80% of images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	67.4563	68.6996	69.17	69.5565
	Precision	68.2	69.6	70.2	70.3
	Recall	67.5	68.7	69.2	69.6
	F-measure	67.1	68.5	69.0	69.3
Correlation Ranking Filter	Accuracy	68.078	68.6324	69.3716	69.6237
	Precision	68.6	69.5	70.3	70.5
	Recall	68.1	68.6	69.4	69.6
	F-measure	67.7	68.4	69.1	69.4
Gain Ratio feature evaluator	Accuracy	67.7083	68.666	69.2036	69.5565
	Precision	68.5	69.5	70.2	70.3
	Recall	67.7	68.7	69.2	69.6
	F-measure	67.4	68.4	69.0	69.3
Information Gain Ranking Filter	Accuracy	67.6747	68.582	69.2204	69.506
	Precision	68.5	69.5	70.2	70.4
	Recall	67.7	68.6	69.2	69.5
	F-measure	67.4	68.4	69.0	69.3
OneR feature evaluator	Accuracy	67.8091	68.834	69.3884	69.6405
	Precision	68.5	69.5	70.3	70.5
	Recall	67.8	68.8	69.4	69.6
	F-measure	67.5	68.5	69.1	69.4
ReliefF Ranking Filter	Accuracy	68.1788	69.002	69.4724	69.2204
	Precision	69.0	70.0	70.5	70.1
	Recall	68.2	69.0	69.5	69.2
	F-measure	67.9	68.8	69.3	69.0
Symmetrical Uncertainty Ranking Filter	Accuracy	67.6579	68.6828	69.1868	69.4556
	Precision	68.5	69.5	70.2	70.3
	Recall	67.7	68.7	69.2	69.5
	F-measure	67.4	68.4	69.0	69.2
No Filter	Accuracy	69.5901			
	Precision	70.4			
	Recall	69.6			
	F-measure	69.3			

VGG16 FC6 with 30 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	66.0	67.5249	68.1855	68.2398
	Precision	69.3	70.6	71.3	71.4
	Recall	66.0	67.5	68.2	68.2
	F-measure	66.7	68.2	68.8	68.9
Correlation Ranking Filter	Accuracy	65.638	67.0679	68.1267	68.3439
	Precision	68.7	70.2	71.2	71.5
	Recall	65.6	67.1	68.1	68.3
	F-measure	66.3	67.7	68.8	69.0
Gain Ratio feature evaluator	Accuracy	65.914	67.5249	68.1855	68.2398
	Precision	69.2	70.6	71.3	71.4
	Recall	65.9	67.5	68.2	68.2
	F-measure	66.6	68.2	68.8	68.9
Information Gain Ranking Filter	Accuracy	66.1493	67.5249	68.1855	68.2398
	Precision	69.4	70.6	71.3	71.4
	Recall	66.1	67.5	68.2	68.2
	F-measure	66.9	68.2	68.8	68.9
OneR feature evaluator	Accuracy	66.5792	67.629	68.1946	68.3258
	Precision	69.7	70.8	71.5	71.5
	Recall	66.6	67.6	68.2	68.3
	F-measure	67.2	68.3	68.9	69.0
ReliefF Ranking Filter	Accuracy	65.086	66.3665	67.6923	68.2624
	Precision	68.4	69.6	71.0	71.5
	Recall	65.1	66.4	67.7	68.3
	F-measure	65.7	67.0	68.4	69.0
Symmetrical Uncertainty Ranking Filter	Accuracy	65.9729	67.5249	68.1855	68.2398
	Precision	69.3	70.6	71.3	71.4
	Recall	66.0	67.5	68.2	68.2
	F-measure	66.7	68.2	68.8	68.9
No Filter	Accuracy	68.4796			
	Precision	71.7			
	Recall	68.5			
	F-measure	69.2			

VGG16 FC6 with 60 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	70.7143	71.9764	72.5173	72.975
	Precision	75.0	76.1	76.6	76.8
	Recall	70.7	72.0	72.5	73.0
	F-measure	71.7	72.9	73.5	73.9
Correlation Ranking Filter	Accuracy	70.6241	71.9209	72.6006	72.975
	Precision	74.8	76.0	76.6	76.8
	Recall	70.6	71.9	72.6	73.0
	F-measure	71.6	72.9	73.5	73.9
Gain Ratio feature evaluator	Accuracy	70.7975	72.2261	72.5659	72.9057
	Precision	75.1	76.2	76.6	76.8
	Recall	70.8	72.2	72.6	72.9
	F-measure	71.8	73.1	73.5	73.8
Information Gain Ranking Filter	Accuracy	70.8322	71.8377	72.552	73.0236
	Precision	74.9	75.9	76.6	76.9
	Recall	70.8	71.8	72.6	73.0
	F-measure	71.7	72.7	73.5	73.9
OneR feature evaluator	Accuracy	70.9778	71.9972	72.5936	72.9958
	Precision	75.1	76.2	76.8	77.1
	Recall	71.0	72.0	72.6	73.0
	F-measure	71.9	73.0	73.6	73.9
ReliefF Ranking Filter	Accuracy	70.0277	71.5673	72.4272	72.8918
	Precision	74.6	75.8	76.4	76.8
	Recall	70.0	71.6	72.4	72.9
	F-measure	71.1	72.5	73.4	73.8
Symmetrical Uncertainty Ranking Filter	Accuracy	70.8807	71.8655	72.6283	73.0444
	Precision	75.0	75.9	76.6	76.9
	Recall	70.9	71.9	72.6	73.0
	F-measure	71.8	72.8	73.6	73.9
No Filter	Accuracy	73.2455			
	Precision	77.1			
	Recall	73.2			
	F-measure	74.1			

VGG16 FC6 with 80% of images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	74.5464	75.4032	76.1425	76.6297
	Precision	75.1	76.1	76.9	77.4
	Recall	74.5	75.4	76.1	76.6
	F-measure	74.3	75.2	76.0	76.5
Correlation Ranking Filter	Accuracy	74.5128	75.2016	76.0585	76.4953
	Precision	75.1	75.8	76,8	77.2
	Recall	74.5	75.2	76,1	76.5
	F-measure	74.2	75.0	75,9	76.3
Gain Ratio feature evaluator	Accuracy	74.1431	75.4872	76.2097	76.6297
	Precision	74.7	76.3	77,0	77.3
	Recall	74.1	75.5	76,2	76.6
	F-measure	73.9	75.4	76,0	76.4
Information Gain Ranking Filter	Accuracy	74.4456	75.5376	76.1089	76.5625
	Precision	75.0	76.1	76,8	77.2
	Recall	74.4	75.5	76,1	76.6
	F-measure	74.2	75.3	75,9	76.4
OneR feature evaluator	Accuracy	74.7984	75.3024	76.0921	76.5289
	Precision	75.1	75.9	76,8	77,2
	Recall	74.8	75.3	76,1	76,5
	F-measure	74.5	75.1	75,9	76,4
ReliefF Ranking Filter	Accuracy	74.1767	74.9328	76.0081	76.3273
	Precision	74.9	75.8	76,8	77,1
	Recall	74.2	74.9	76,0	76,3
	F-measure	74.0	74.8	75,8	76,2
Symmetrical Uncertainty Ranking Filter	Accuracy	74.5464	75.5376	76.0417	76.5625
	Precision	75.2	76.2	76,8	77,3
	Recall	74.5	75.5	76,0	76,6
	F-measure	74.3	75.3	75,9	76,4
No Filter	Accuracy	76.5289			
	Precision	77.3			
	Recall	76.5			
	F-measure	76.4			

VGG16 FC7 with 30 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	66.8869	68.1403	68.6018	68.7059
	Precision	70.0	71.1	71.5	71.6
	Recall	66.9	68.1	68.6	68.7
	F-measure	67.6	68.8	69.2	69.4
Correlation Ranking Filter	Accuracy	66.5837	67.8552	68.4253	68.6787
	Precision	69.8	70.8	71.5	71.7
	Recall	66.6	67.9	68.4	68.7
	F-measure	67.3	68.5	69.1	69.3
Gain Ratio feature evaluator	Accuracy	66.7195	68.009	68.6018	68.7059
	Precision	69.9	71.0	71.5	71.6
	Recall	66.7	68.0	68.6	68.7
	F-measure	67.4	68.7	69.2	69.4
Information Gain Ranking Filter	Accuracy	66.8462	68.1357	68.6018	68.7059
	Precision	70.0	71.1	71.5	71.6
	Recall	66.8	68.1	68.6	68.7
	F-measure	67.5	68.8	69.2	69.4
OneR feature evaluator	Accuracy	67.4977	68.2489	68.81	68.6787
	Precision	70.5	71.2	71.8	71.6
	Recall	67.5	68.2	68.8	68.7
	F-measure	68.1	68.9	69.4	69.3
ReliefF Ranking Filter	Accuracy	66.3167	67.4796	68.2081	68.457
	Precision	69.5	70.5	71.2	71.5
	Recall	66.3	67.5	68.2	68.5
	F-measure	67.0	68.1	68.8	69.1
Symmetrical Uncertainty Ranking Filter	Accuracy	67.00	68.00	68.6018	68.7059
	Precision	70.1	71.0	71.5	71.6
	Recall	67.0	68.0	68.6	68.7
	F-measure	67.7	68.6	69.2	69.4
No Filter	Accuracy	68.7557			
	Precision	71.7			
	Recall	68.8			
	F-measure	69.4			

VGG16 FC7 with 60 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	71.6019	72.5589	73.19	73.3703
	Precision	75.4	76.0	76.6	76.7
	Recall	71.6	72.6	73.2	73.4
	F-measure	72.5	73.4	74.0	74.1
Correlation Ranking Filter	Accuracy	71.6089	72.8918	73.1553	73.4327
	Precision	75.3	76.3	76.6	76.8
	Recall	71.6	72.9	73.2	73.4
	F-measure	72.5	73.7	74.0	74.2
Gain Ratio feature evaluator	Accuracy	71.4979	72.5381	73.0721	73.4119
	Precision	75.2	76.0	76.5	76.7
	Recall	71.5	72.5	73.1	73.4
	F-measure	72.4	73.4	73.9	74.2
Information Gain Ranking Filter	Accuracy	71.6852	72.663	73.1484	73.3634
	Precision	75.5	76.1	76.5	76.6
	Recall	71.7	72.7	73.1	73.4
	F-measure	72.6	73.5	73.9	74.1
OneR feature evaluator	Accuracy	72.1914	72.9334	73.2316	73.4743
	Precision	75.7	76.2	76.5	76.8
	Recall	72.2	72.9	73.2	73.5
	F-measure	73.0	73.7	74.0	74.3
ReliefF Ranking Filter	Accuracy	71.5187	72.3925	72.9473	73.0999
	Precision	75.4	76.6	76.3	76.5
	Recall	71.5	72.4	72.9	73.1
	F-measure	72.4	73.2	73.7	73.9
Symmetrical Uncertainty Ranking Filter	Accuracy	71.6574	72.5659	73.1831	73.3356
	Precision	75.4	76.1	76.6	76.6
	Recall	71.7	72.6	73.2	73.3
	F-measure	72.6	73.4	74.0	74.1
No Filter	Accuracy	73.4535			
	Precision	76.8			
	Recall	73.5			
	F-measure	74.2			

VGG16 FC7 with 80% of images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	75.252	75.6888	76.1257	76.2433
	Precision	76.0	76.0	76.8	76.6
	Recall	75.3	75.7	76.1	76.2
	F-measure	75.1	75.3	75.9	76.0
Correlation Ranking Filter	Accuracy	75.1176	75.5208	76.0753	76.3105
	Precision	75.8	76.3	76.6	76.8
	Recall	75.1	75.5	76.1	76.3
	F-measure	74.9	75.3	75.8	76.0
Gain Ratio feature evaluator	Accuracy	74.2608	75.3864	76.0417	76.1593
	Precision	74.9	76.1	76.6	76.6
	Recall	74.3	75.4	76.0	76.2
	F-measure	74.0	75.2	75.8	75.9
Information Gain Ranking Filter	Accuracy	75.3528	75.4872	75.8905	76.3777
	Precision	76.0	76.2	76.5	76.8
	Recall	75.4	75.5	75.9	76.4
	F-measure	75.1	75.3	75.7	76.1
OneR feature evaluator	Accuracy	75.1512	75.4536	76.0249	76.1089
	Precision	75.8	75.9	76.5	76.6
	Recall	75.2	75.5	76.0	76.1
	F-measure	74.9	75.1	75.8	75.8
ReliefF Ranking Filter	Accuracy	74.4624	75.2016	75.756	76.1593
	Precision	75.2	75.9	76.2	76.6
	Recall	74.5	75.2	75.8	76.2
	F-measure	74.2	75.0	75.5	75.9
Symmetrical Uncertainty Ranking Filter	Accuracy	75.3024	75.5712	75.8905	76.3105
	Precision	76.0	76.3	76.5	76,7
	Recall	75.3	75.6	75.9	76,3
	F-measure	75.1	75.4	75.7	76,0
No Filter	Accuracy	76.4785			
	Precision	77.0			
	Recall	76.5			
	F-measure	76.2			

VGG19 FC6 with 30 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	66.1086	67.5611	68.4434	68.9231
	Precision	69.8	71.1	71.8	72.2
	Recall	66.1	67.6	68.4	68.9
	F-measure	67.0	68.4	69.2	69.7
Correlation Ranking Filter	Accuracy	65.9593	67.4525	68.3665	68.6335
	Precision	69.6	70.8	71.7	71.9
	Recall	66.0	67.5	68.4	68.6
	F-measure	66.8	68.2	69.1	69.4
Gain Ratio feature evaluator	Accuracy	66.1086	67.5611	68.4434	68.9231
	Precision	69.8	71.1	71.8	72.2
	Recall	66.1	67.6	68.4	68.9
	F-measure	67.0	68.4	69.2	69.7
Information Gain Ranking Filter	Accuracy	66.1086	67.5611	68.4434	68.9231
	Precision	69.8	71.1	71.8	72.2
	Recall	66.1	67.6	68.4	68.9
	F-measure	67.0	68.4	69.2	69.7
OneR feature evaluator	Accuracy	66.8824	67.9186	68.4751	68.81
	Precision	70.2	71.3	71.7	72.1
	Recall	66.9	67.9	68.5	68.8
	F-measure	67.6	68.7	69.2	69.6
ReliefF Ranking Filter	Accuracy	65.2941	66.914	68.19	68.4706
	Precision	69.2	70.6	71.7	71.8
	Recall	65.3	66.9	68.2	68.5
	F-measure	66.2	67.8	69.0	69.2
Symmetrical Uncertainty Ranking Filter	Accuracy	66.1086	67.5611	68.4434	68.9231
	Precision	69.8	71.1	71.8	72.2
	Recall	66.1	67.6	68.4	68.9
	F-measure	67.0	68.4	69.2	69.7
No Filter	Accuracy	68.8914			
	Precision	72.2			
	Recall	68.9			
	F-measure	69.7			

VGG19 FC6 with 60 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	71.1096	72.1567	72.9265	73.2663
	Precision	75.4	76.2	76.8	77.1
	Recall	71.1	72.2	72.9	73.3
	F-measure	72.1	73.1	73.8	74.1
Correlation Ranking Filter	Accuracy	70.9778	72.0666	72.767	73.3426
	Precision	75.2	76.3	76.7	77.2
	Recall	71.0	72.1	72.8	73.3
	F-measure	72.0	73.0	73.7	74.2
Gain Ratio feature evaluator	Accuracy	70.957	72.0666	73.0374	73.2663
	Precision	75.4	76.2	76.9	77.1
	Recall	71.0	72.1	73.0	73.3
	F-measure	72.0	73.0	73.9	74.1
Information Gain Ranking Filter	Accuracy	70.8183	72.1775	72.9265	73.301
	Precision	75.1	76.3	76.8	77.1
	Recall	70.8	72.2	72.9	73.3
	F-measure	71.8	73.1	73.8	74.2
OneR feature evaluator	Accuracy	71.6852	72.448	72.9612	73.2108
	Precision	75.5	76.5	77.0	77.1
	Recall	71.7	72.4	73.0	73.2
	F-measure	72.6	73.4	73.9	74.1
ReliefF Ranking Filter	Accuracy	70.3467	71.5534	72.4619	73.0929
	Precision	74.7	75.8	76.5	76.9
	Recall	70.3	71.6	72.5	73.1
	F-measure	71.4	72.6	73.4	74.0
Symmetrical Uncertainty Ranking Filter	Accuracy	70.9015	72.0943	72.8779	73.301
	Precision	75.2	76.2	76.8	77.1
	Recall	70.9	72.1	72.9	73.3
	F-measure	71.9	73.0	73.8	74.2
No Filter	Accuracy	73.5714			
	Precision	77.3			
	Recall	73.6			
	F-measure	74.4			

VGG19 FC6 with 80% of images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	74.8488	75.5712	76.3105	76.5961
	Precision	75.6	76.3	76.8	77.2
	Recall	74.8	75.6	76.3	76.6
	F-measure	74.7	75.4	76.1	76.4
Correlation Ranking Filter	Accuracy	74.6304	75.6552	76.0921	76.4449
	Precision	75.4	76.4	76,6	77.1
	Recall	74.6	75.7	76,1	76.4
	F-measure	74.5	75.5	75,9	76.3
Gain Ratio feature evaluator	Accuracy	74.916	75.5208	76.3945	76.5793
	Precision	75.6	76.3	77,0	77.1
	Recall	74.9	75.5	76,4	76.6
	F-measure	74.7	75.3	76,2	76.4
Information Gain Ranking Filter	Accuracy	74.7984	75.6384	76.1929	76.5625
	Precision	75.7	76.4	76,7	77,2
	Recall	74.8	75.6	76,2	76,6
	F-measure	74.6	75.5	75,9	76,4
OneR feature evaluator	Accuracy	75.168	75.9745	76.6969	76.7137
	Precision	75.7	76.7	77,4	77,3
	Recall	75.2	76.0	76,7	76,7
	F-measure	74.8	75.8	76,5	76,5
ReliefF Ranking Filter	Accuracy	74.2103	75.5712	76.1593	76.5121
	Precision	75.0	76.4	76,8	77,2
	Recall	74.2	75.6	76,2	76,5
	F-measure	74.0	75.3	75,9	76,3
Symmetrical Uncertainty Ranking Filter	Accuracy	74.748	75.42	76.2097	76.3945
	Precision	75.7	76.1	76,7	77,0
	Recall	74.7	75.4	76,2	76,4
	F-measure	74.6	75.2	76,0	76,2
No Filter	Accuracy	76.4953			
	Precision	77.1			
	Recall	76.5			
	F-measure	76.3			

VGG19 FC7 with 30 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	67.7466	68.724	69.1357	69.267
	Precision	70.5	71.5	71.9	71.9
	Recall	67.7	68.7	69.1	69.3
	F-measure	68.3	69.3	69.7	69.9
Correlation Ranking Filter	Accuracy	67.362	68.2398	69.0	69.2172
	Precision	70.0	70.9	71.7	71.9
	Recall	67.4	68.2	69.0	69.2
	F-measure	67.9	68.8	69.6	69.8
Gain Ratio feature evaluator	Accuracy	67.543	68.724	69.1357	69.267
	Precision	70.3	71.5	71.9	71.9
	Recall	67.5	68.7	69.1	69.3
	F-measure	68.1	69.3	69.7	69.9
Information Gain Ranking Filter	Accuracy	67.5249	68.724	69.1357	69.267
	Precision	70.2	71.5	71.9	71.9
	Recall	67.5	68.7	69.1	69.3
	F-measure	68.1	69.3	69.7	69.9
OneR feature evaluator	Accuracy	67.8552	68.6018	69.0226	69.2172
	Precision	70.7	71.4	71.8	72.0
	Recall	67.9	68.6	69.0	69.2
	F-measure	68.5	69.2	69.6	69.8
ReliefF Ranking Filter	Accuracy	66.7828	68.0452	68.6244	69.0181
	Precision	69.6	70.9	71.5	71.8
	Recall	66.8	68.0	68.6	69.0
	F-measure	67.4	68.7	69.3	69.7
Symmetrical Uncertainty Ranking Filter	Accuracy	67.6787	68.724	69.1357	69.267
	Precision	70.2	71.5	71.9	71.9
	Recall	67.7	68.7	69.1	69.3
	F-measure	68.2	69.3	69.7	69.9
No Filter	Accuracy	69.3348			
	Precision	72.0			
	Recall	69.3			
	F-measure	69.9			

VGG19 FC7 with 60 images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	71.491	72.2677	73.1692	73.0791
	Precision	75.6	76.4	77.1	76.9
	Recall	71.5	72.3	73.2	73.1
	F-measure	72.5	73.2	74.1	73.9
Correlation Ranking Filter	Accuracy	71.7892	72.4549	72.9542	73.2108
	Precision	76.1	76.5	76.9	76.9
	Recall	71.8	72.5	73.0	73.2
	F-measure	72.8	73.4	73.9	74.1
Gain Ratio feature evaluator	Accuracy	71.3454	72.2469	73.1761	73.0236
	Precision	75.7	76.3	77.1	76.9
	Recall	71.3	72.2	73.2	73.0
	F-measure	72.4	73.2	74.1	73.9
Information Gain Ranking Filter	Accuracy	71.6436	72.1983	72.975	73.0721
	Precision	75.9	76.3	76.9	76.9
	Recall	71.6	72.2	73.0	73.1
	F-measure	72.6	73.1	73.9	73.9
OneR feature evaluator	Accuracy	71.7614	72.6838	73.0791	73.1831
	Precision	76.0	76.7	76.9	76.9
	Recall	71.8	72.7	73.1	73.2
	F-measure	72.7	73.6	73.9	74.0
ReliefF Ranking Filter	Accuracy	71.81	72.3994	72.975	73.3842
	Precision	75.8	76.2	76.8	77.1
	Recall	71.8	72.4	73.0	73.4
	F-measure	72.7	73.2	73.8	74.2
Symmetrical Uncertainty Ranking Filter	Accuracy	71.6436	72.2885	73.0513	73.0652
	Precision	75.9	76.4	77.0	76.9
	Recall	71.6	72.3	73.1	73.1
	F-measure	72.7	73.2	74.0	73.9
No Filter	Accuracy	73.3495			
	Precision	77.0			
	Recall	73.3			
	F-measure	74.1			

VGG19 FC7 with 80% of images per class					
Methods	Metrics	**Number of Selected Features**			
		500	**1000**	**2000**	**3000**
Chi-squared Ranking Filter	Accuracy	75.084	75.4704	76.4113	76.5625
	Precision	75.4	76.2	76.8	76.9
	Recall	75.1	75.5	76.4	76.6
	F-measure	74.7	75.3	76.1	76.2
Correlation Ranking Filter	Accuracy	75.1512	76.0585	76.0753	76.4953
	Precision	75.6	76.3	76,4	76.8
	Recall	75.2	76.1	76,1	76.5
	F-measure	74.8	75.7	75,7	76.1
Gain Ratio feature evaluator	Accuracy	75.1512	75.756	76.4113	76.6297
	Precision	75.6	76.0	76,8	77.0
	Recall	75.2	75.8	76,4	76.6
	F-measure	74.8	75.4	76,1	76.3
Information Gain Ranking Filter	Accuracy	75.0168	75.8233	76.4617	76.5457
	Precision	75.4	76.2	76,9	76,9
	Recall	75.0	75.8	76,5	76,5
	F-measure	74.7	75.4	76,1	76,2
OneR feature evaluator	Accuracy	75.756	75.9409	76.5289	76.5961
	Precision	76.3	76.3	76,8	76,9
	Recall	75.8	75.9	76,5	76,6
	F-measure	75.4	75.6	76,2	76,2
ReliefF Ranking Filter	Accuracy	75.084	75.7056	76.2937	76.2769
	Precision	75.5	76.1	76,7	76.7
	Recall	75.1	75.7	76,3	76.3
	F-measure	74.8	75.4	76,0	76.0
Symmetrical Uncertainty Ranking Filter	Accuracy	74.9496	75.672	76.4281	76.5121
	Precision	75.3	76.0	76.8	76.9
	Recall	74.9	75.7	76.4	76.5
	F-measure	74.6	75.3	76.1	76.2
No Filter	Accuracy	76.3441			
	Precision	76.7			
	Recall	76.3			
	F-measure	76.0			

APPENDIX D

airplanes	car-tire	electric-guitar	greyhound	knife	pci-card	snail	theodolite
ak47	cartman	elephant	guitar-pick	ladder	penguin	snake	toad
american-flag	cd	elk	hamburger	laptop	people	sneaker	toaster
backpack	centipede	ewer	hammock	lathe	pez-dispenser	snowmobile	tomato
baseball-bat	cereal-box	eyeglasses	harmonica	leopards	photocopier	soccer-ball	tombstone
baseball-glove	chandelier	faces	harp	license-plate	picnic-table	socks	top-hat
basketball-hoop	chess-board	fern	harpsichord	light-house	playing-card	soda-can	touring-bike
bat	chimp	fighter-jet	hawksbill	lightbulb	porcupine	spaghetti	tower-pisa
bathtub	chopsticks	fire-extinguisher	head-phones	lightning	pram	speedboat	traffic-light
bear	cockroach	fire-hydrant	helicopter	llama	praying-mantis	spider	treadmill
beer-mug	coffee-mug	fire-truck	hibiscus	mailbox	pyramid	spoon	triceratops
billiards	coffin	fireworks	homer-simpson	mandolin	raccoon	stained-glass	tricycle
binoculars	coin	flashlight	horse	mars	radio-telescope	starfish	trilobite
birdbath	comet	floppy-disk	horseshoe-crab	mattress	rainbow	steering-wheel	tripod
blimp	computer-keyword	football-helmet	hot-air-balloon	megaphone	refrigerator	stirrups	tuning-fork
bonsai	computer-monitor	french-horn	hot-dog	menorah	revolver	sunflower	tweezer
boom-box	computer-mouse	fried-egg	hot-tub	microscope	rifle	superman	umbrella
bowling-ball	conch	frisbee	hourglass	microwave	rotary-phone	sushi	unicorn
bowling-pin	cormorant	frog	house-fly	minaret	roulette-wheel	swan	vcr
boxing-glove	covered-wagon	frying-pan	human-skeleton	minotaur	saddle	swiss-army-knife	video-projector
brain	cowboy-hat	galaxy	hummingbird	motorbikes	saturn	sword	washing-machine
breadmaker	crab	gas-pump	ibis	mountain-bike	school-bus	syringe	watch
buddha	desk-globe	giraffe	ice-cream-cone	mushroom	scorpion	t-shirt	waterfall
bulldozer	diamond-ring	goat	iguana	mussels	screwdriver	tambourine	watermelon
butterfly	dice	golden-gate-bridge	ipod	necktie	segway	teapot	welding-mask
cactus	dog	goldfish	iris	octopus	self-propelled-lawn-mower	teddy-bear	wheelbarrow
cake	dolphin	golf-ball	jesus-christ	ostrich	sextant	teepee	windmill
calculator	doorknob	goose	joy-stick	owl	sheet-music	telephone-box	wine-bottle
camel	drinking-straw	gorilla	kangaroo	palm-pilot	skateboard	tennis-ball	xylophone
cannon	duck	grand-piano	kayak	palm-tree	skunk	tennis-court	yarmulke
canoe	dumb-bell	grapes	ketch	paper-shredder	skyscraper	tennis-racket	yo-yo
car-side	eiffel-tower	grasshopper	killer-whale	paperclip	smokestack	tennis-shoes	zebra