2017

On Designing Deep Learning Approaches for Classification of Football Jersey Images in the Wild

Rohitha Reddy Matta

On Designing Deep Learning Approaches for Classification of Football Jersey Images in the Wild

Rohitha Reddy Matta

Thesis submitted to the

Benjamin M. Statler College of Engineering and Mineral Resources

at West Virginia University

in partial fulfillment of the requirements

for the degree of

Master of Science

in

Electrical Engineering

Thirimachos Bourlai, Ph.D., Chair

Matthew Valenti, Ph.D.

Jeremy Dawson, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2017

Keywords: Jersey Classification, Convolutional Neural Networks, Classification, Media application, Image Quality Assessment

Abstract

On Designing Deep Learning Approaches for Classification of Football Jersey Images in the Wild

Rohitha Reddy Matta

Internet shopping has spread wide and into social networking. Someone may want to buy a shirt, accessories, etc., in a random picture or a streaming video. In this thesis, the problem of automatic classification was taken upon, constraining the target to jerseys in the wild, assuming the object is detected.

A dataset of 7,840 jersey images, namely the JerseyXIV is created, containing images of 14 categories of various football jersey types (Home and Alternate) belonging to 10 teams of 2015 Big 12 Conference football season. The quality of images varies in terms of pose, standoff distance, level of occlusion and illumination. Due to copyright restrictions on certain images, unaltered original images with appropriate credits can be provided upon request.

While various conventional and deep learning based classification approaches were empirically designed, optimized and tested, a solution that resulted in the highest accuracy in terms of classification was achieved by a train-time fused Convolutional Neural Network (CNN) architecture, namely CNN-F, with 92.61% accuracy. The final solution combines three different CNNs through score level average fusion achieving 96.90% test accuracy. To test these trained CNN models on a larger, application oriented scale, a video dataset is created, which may present an addition of higher rate of occlusion and elements of transmission noise. It consists of 14 videos, one for each class, totaling to 3,584 frames, with 2,188 frames containing the object of interest. With manual detection, the score level average fusion has achieved the highest classification accuracy of 81.31%.

In addition, three Image Quality Assessment techniques were tested to assess the drop in accuracy of the average-fusion method on the video dataset. The Natural Image Quality Evaluator (NIQE) index by Bovik et al. with a threshold of 0.40 on input images improved the test accuracy of the average fusion model on the video dataset to 86.36% by removing the low quality input images before it reaches the CNN.

The thesis concludes that the recommended solution for the classification is composed of data augmentation and fusion of networks, while for application of trained models on videos, an image quality metric would aid in performance increase with a trade-off in loss of input data.

I dedicate my thesis to my family

Acknowledgments

Firstly, I would like to express my deepest appreciation and gratitude to my advisor and committee chair, Dr. Thirimachos Bourlai for giving me the opportunity to work on this project. His knowledge, dedication, encouragement and hardworking nature have inspired me in many ways. I consider it as a privilege to have him as my advisor throughout my work. I also wish to express my sincere thanks to the members of my committee, Dr. Matthew Valenti and Dr. Jeremy Dawson for accepting to serve on my committee.

I would like to thank the project sponsors - Jonathan Ohligher, CEO, and rest of the team from Veepio, for providing the opportunity. I am deeply grateful for their immense support in providing resources, GPUs, office space and the much needed industrial exposure. Project number: 10020139.1.1006928R and Title: "Object Tracking on Unconstrained Videos"

I would like to thank my brother Temujin Reddy Matta for his unparallel support and guidance, and my family members who always believed in me. I would not have made it this far without their support.

I would like to extend my thanks to my lab mates Neeru and Michael for their suggestions, co-operation and support.

Finally, I would like to express my gratitude to all who have given me guidance, encouragement and inspiration during my work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

These days the internet and social networking have become a part of human life. Shopping online has become more convenient and reachable than ever by being just a click away. Veepio, the sponsors of this project, has introduced the idea of interactive shopping in random pictures and videos through their mobile app. In this thesis, automatic classification of objects through machine learning is worked upon considering the object is detected. Since there is a wide variety of apparels, accessories, etc., available in real world, this work selected jersey images in the wild as the target since it provides different challenges in classifying objects in real time videos or pictures.

The publicly available upper body clothing datasets, such as, Deep Fashion [5] and [6] have 15 and 50 categories respectively. They provide images which are taken mostly under controlled conditions of illumination and pose containing complete apparel information. Even though the JerseyXIV dataset has lesser categories and images, it presents high amount of variation in terms of pose, illumination and occlusions due to the football players being in constant motion. This problem is also applicable as a scenario useful for surveillance applications when the focus can be to detect one or more subjects within a crowded scene wearing a specific T-shirt, shirt or jacket.

In order to generate the JerseyXIV dataset for testing upper body clothing (jersey) classifi-

cation approaches, jersey images of American Football are selected, which is one of the most popular sports in the United States [7]. All available teams of the Big-XII-2015 Conference, which is a member of the NCAA (National Collegiate Athletic Association) Division-1 are taken as the categories. Specifically, the dataset is composed of jersey images from 10 college teams collected from online sources. The dataset is a collection of frontal and off-angle images of jerseys, the majority of which are of players during play in an actual season game. An example of the dataset can be seen in (see Fig. 1.1).

Figure 1.1: Example football jersey images in the wild from the JerseyXIV dataset composed of 14 different classes. One image from each category is selected to show the variation as a whole. From top left to right: Baylor Alternate, Baylor Home, Iowa Home, Kansas Jayhwaks Home, Kansas Wildcats Home, Oklahoma Cowboys Alternate, Oklahoma Cowboys Home. From bottom left to right: Oklahoma Sooners Home, Texas Tech Home, Texas Horned Frogs Alternate, Texas Horned Frogs Home, Texas Longhorns Home, West Virginia Alternate, West Virginia Home. (jerseys as of 2015).

Recently, deep neural networks are used as a preferred method for challenging classification problems such as the one focused in this work, since it demonstrated good results on various benchmark datasets [8][9]. Prior to the recent focus in the usage of deep learning algorithms, various conventional methods were and are still used for comparison. However, such approaches can work better with relatively good quality images. In addition, in such methods, several modules of the classification approach may depend on the operators manual intervention (e.g., manual object detection and background subtraction). In contrast, deep learning based approaches can usually work better with challenging datasets [8]. They include self-adaptive algorithms, which do not need any prior knowledge and can approximate any function [10]. In particular, Convolutional Neural Networks have been advancing to be the most reliable approach when designing object recognition algorithms [8] [9]. CNNs seem to achieve good

results on benchmark data sets of various objects. In this thesis, the robustness of the CNNs to classify the images of novel data is also investigated.

1.2 Problem Statement

The dataset consists of 14 categories with 1 or 2 sub-categories (Home and Alternate) per team. These are a few challenging factors in the image and video data set collected (see Fig. 1.2)

(i) The images demonstrate significant variations in terms of pose, standoff distance, illumination and occlusion.

(ii) The design of the home and alternate jerseys does not vary a lot.

(iii) The size and location of the jersey number and text varies.

(iv) The jersey images available in the dataset coming from different teams may have the same jersey number.

(v) The video dataset which is created only for testing the models, show occlusion, blurness caused by camera movement, cluttered scene and losing object.

Figure 1.2: An example showing the challenges in the image and video dataset, namely, illumination, occlusion, pose, camera movement, losing object and cluttered scene

The goal is to classify the 14 classes of the Jersey XIV dataset with high accuracy, by studying and experimenting with various classification methods - such as HOG with an SVM classifier, Bag of Words and various architectures of Convolutional neural networks, such as

Inception-v3, Inception-ResNet-v2 modified models and transfer learning. The learned models are also tested on a novel video dataset to test the robustness. An image quality assessment study is also done in order to find in which conditions the system fails in classifying the images and thereby improving the classification performance.

1.3 Contribution of Thesis

In this work, an experimental study is conducted for the classification of the Jersey XIV dataset and is established that the fusion of networks in addition with data augmentation, is superior to other models tested. The contribution of the thesis is three-fold.

First, a dataset of 7,840 jersey images belonging to 14 categories are collected from various online resources. These images are of 2015 Big XII Conference football season. This dataset is augmented by 5 times for training deep networks.

Second, two main solutions, namely, an Inception-ResNet-v2 based modified network (CNN-IR) and a train-time fusion network (CNN-F) are proposed, which achieved 91.42% and 92.61% test accuracy on image dataset respectively. The score level average fusion of the models has achieved the best, i.e., 96.90%. These proposed solutions are compared with other conventional empirically optimized solutions such as Bag of Words (BOW) and Histogram of Oriented Gradients (HOG) with a Support Vector Machine (SVM).

Third, the trained models are tested on a larger scale video dataset, which consists of a higher rate of occlusions and possible transmission noises. The video dataset consists of 14 videos for each of 14 classes, totaling to 3,584 frames, of which 2,188 frames containing the object of interest (jersey). The objects of interest are manually annotated with bounding boxes to mimic an object detector. The best test accuracy of 81.31% is achieved by the score level average fusion of three models.

In addition, no-reference image quality assessment of the 2,188 images of the video dataset using Subjective IQA and objective assessments: Spatial-Spectral Entropy-based Quality (SSEQ) index [11] and Natural Image Quality Evaluator (NIQE) [12] are evaluated for increased performance. NIQE is found to be the better IQA method to apply a threshold for discarding distorted

input images, increasing the video test accuracy of score level average-fusion to 86.32% accuracy.

1.4 Thesis Organization

The remainder of the thesis is organized as follows:

- Chapter 2 describes existing work related to image classification using conventional, convolutional neural networks, transfer learning methods and Image quality assessment.
- Chapter 3 describes the overall experimental setup, dataset collection and data augmentation. It is divided into three parts, Part-1 presents the methodology of Bag of Words (BOW), Histogram of oriented gradient (HOG) with Support Vector Machine (SVM). Part-2 presents the developed convolutional neural network architectures - Initial baseline CNN network (CNN-s for grayscale input and CNN-s-r for RGB input), Inception-v3 retrained model (Inception-v3-R+), Modified InceptionResNet-v2 (CNN-IR) and train-time fusion networks (CNN-F) explained in detail. The score level Average-fusion of the above mentioned models is also explained here. Part-3 discusses the evaluation of models on video dataset and no-reference image quality assessment methods.
- Chapter 4 presents the experimental evaluations and results of the methods described in Chapter-3's Part-1, Part-2 and Part-3.
- Chapter 5 concludes with thesis contributions, limitations and future work.

Chapter 2

Related Work

2.1 Background and Conventional Classification Methods

Pattern Recognition has been an active and interesting problem to solve in the field of computer vision and artificial intelligence. Many systems and methods were developed for decades to make the recognition systems accurate enough, closing the gap between automated computer vision based methods and human precision.

Classification is a supervised learning method of Pattern Recognition algorithms, where the algorithm maps the input data to a category belonging to certain predefined categories. Feature extraction methods are important in capturing the required features of the input and in training a classifier. The earliest form of features used are the shape, region, geometric structure or the implicit model of the shape of the object [13] [14]. These features cannot be extended for complex real-world shapes as it increases the computational cost. To address this problem, appearance based and feature based methods were introduced and further used to perform classification. Appearance based methods use templates of the objects to perform recognition, while feature based methods are based on features extracted such as edges, patches, corners, shapes, and active contours. Appearance based methods are invariant to image scale and rotation, but require images taken in a controlled environment. This method is not robust to occlusions, unknown backgrounds and illumination conditions.

Feature-based methods were introduced to overcome the disadvantages of appearance based

methods. The feature extraction methods used in this thesis are feature-based. Histogram of Oriented Gradient (HOG) is a feature based descriptor, similar to edge orientation histograms or Scale-Invariant feature transform (SIFT) descriptors, but are computed on a dense grid of uniformly spaced cells and use overlapping local contrast normalizations for improved performance [15]. HOG provides excellent performance relative to other existing feature sets including wavelets [15]. In this work HOG feature based descriptor is used with a Support Vector Machine (SVM) classifier to establish baseline jersey classification performance. However, the method had to be carefully handcrafted for the classification problem and did not result in satisfactory results.

Bag of Visual Words (BOW) or Bag of features (BOF) was introduced for categorizing words of text in natural language processing and in image retrieval. The method creates 'bag of words' consisting of frequency of occurrence of each words and is used as a feature for training a classifier. The method can be applied/extended to recognition, content based image retrieval and detection. For classification purpose, Csukara et al. [16] introduced bag of keypoints method for visual categorization, where bag of keypoints represents a histogram of number of occurrences of particular image pattern in the image. Feature descriptors of image patches are extracted and grouped into clusters by using kmeans clustering to form a set of vocabularies, from which bag of keypoints are extracted and used as feature vector to train a classifier. The paper have shown that the method is robust to background clutter, invariant to affine transformations, occlusion, lighting and intra-class variations. Since the JerseyXIV dataset is subject to the variants mentioned above, BOW is tested in this thesis.

2.2 Deep Neural Networks for Classification

Deep neural networks were introduced instead of the above methods as an alternative and fast learning approach. Deep neural networks extract their own features without much manual intervention from the network operator. Convolutional Neural Networks (CNNs) are a type of deep learning architecture inspired by the function of animal visual cortex [17]. In 1998, Lecun et al. [18] introduced one of the very first convolutional neural networks called LeNet5

for image classification using a gradient based backpropagation algorithm for training. It was trained on 10 class hand-written digit recognition dataset.

In recent years, with the advance in technology and faster training on multiple GPUs, Convolutional neural networks have shown to produce higher accuracy in image classification with a thousand categories, such as, AlexNet by Krizhevsky [8] and VGG net by Simonyan [19] which documented 23.7% top-1 error while the former reported top-1 error of 37.5%. AlexNet is trained on ILSVRC dataset 2010, while VGG on ILSVRC 2014 dataset, each containing 1,000 classes.

Szegedy [3] introduced a combined version of Inception and Residual networks [20], utilizing the deep structure of inception and the additive merging of residual networks. This Inception-ResNet-v2 network is trained on ILSVRC 2012 dataset with 19.9% top-1 error. Even though the computational cost of this network is higher, there is an increase in recognition performance. Based on this network, CNN-IR is constructed.

CNN models with trained weights can also be used as feature extractors. One such interesting deep learning based study was reported by Razavian [21]. They trained an SVM classifier on the features extracted from an Imagenet-trained OverFeat network and have reported state-of-the-art results. Their work also added a mounting evidence that the features extracted from a trained network are very powerful and are transferable to other datasets as well.

While deep learning can be a very powerful tool on its own, it has several limitations including, (i) advanced hardware requirements to accelerate the design process of challenging classification experiments, but also (ii) the need for sufficient data for training. In the latter case, data augmentation may be necessary to improve classification performance. It is used to increase an original, small scale dataset, and when used properly, it can reduce overfitting issues [8].

In this work, the structure of baseline CNN-s network, which is trained on the JerseyIV Image dataset, is built by following the general design principles used in LeNet and AlexNet. The other two proposed networks are based on Inception-ResNetv2. The input image size in all the network architectures is 64×64, in contrast to the generally used higher dimension of 224×224 and 299×299. A dropout layer is used to reduce the overfitting and activations per

tile were increased per layer to generate more distangled features. For transfer learning, publicly available trained network (Inception-v3) proposed by Vanhoucke [9] is used to retrain its last classification layer. The Inception-v3 is originally trained on the Imagenet ILSVRC-2012 dataset that contains 1,000 classes. The originally collected jersey in the wild dataset is also augmented by 5 times with the expectation to improve classification performance. A detailed explanation of the baseline, modified Inception-ResNetv2 and train-time fused networks are presented in the methodology section.

2.3 Image Quality Assessment

The quality characteristics measure the amount of visual degradation in an image. Image Quality Assessment (IQA) is a method of measuring these degradations in the images, which can occur due to the presence of noise, blur, fading, compression, etc. Image quality assessment can be done in two ways: Subjective and Objective. Subjective image quality assessment is done by human observers and is usually time consuming and not always accurate. Objective image quality assessment refers to the automatic prediction of quality of distorted images. An automatic quality assessment method would be more preferable for speed, accuracy and can be embedded into the processing system [22]. The images can be analyzed for distortions - noise, blur, compression and fast fading.

There are three types of Image Quality Assessment (IQA) based on the availability of reference images: full-reference (FR), reduced-reference (RR) and no-reference(NR). The full-reference QA needs an undistorted reference image and is basically finding the similarity index. The reduced-reference IQA is used when the undistorted reference is not fully available, but some of its features as *a priori* information is available. A no-reference or blind QA is employed when a reference image is not available. In this work, the classification is executed in real-time without any reference image and so no-reference image quality assessment methods are employed.

Even though there has been quite a research done under no-reference IQA, many algorithms are only designed for one or two distortions, such as, JPEG-2k compression [23], JPEG

compression [24], JPEG-2K and JPEG compression [25], blur [26] and, blurred and JPEG-2k compression [27]. Each of these methods computes distortions to a specific type.

Overcoming the limited type of distortions as in the above, the following papers have proposed training and learning based approaches. Moorthy and Bovik [28] has proposed a simple blind IQA named BIQI, using NSS features. Then they added two-step framework and proposed Distortion Identification based Image Verity and INtegrity Evaluation (DIIVINE) index, which is based on the theory that the statistical properties of an image changes with the presence of distortions [29]. DIIVINE index uses NSS features in the wavelet domain to predict the quality scores. BLIINDS-II by Saad et al. uses block based discrete cosine transformation to extract NSS features using a fast single-stage framework [30]. Blind/Reference-less Image Spatial QUality Evaluator (BRISQUE) is introduced by Mittal et al. which extracts features in spatial domain achieving superior results than the above methods [31].

Liu and Bovik [11] introduced Spatial-Spectral Entropy-based Quality (SSEQ) Index, which can access five types of distortions, namely, Gaussian blur, white noise, JPEG compression, JPEG-2k compression and fast fading. The model utilizes local spatial and spectral frequency entropies of local patches as features. This method is statistically superior to all the no-reference methods mentioned above and close to human opinions.

All the methods mentioned so far are opinion-aware models. Mittal et al. introduced a first of a kind opinion-unaware and distortion-unaware model which does not require any exposure to distorted images or training on human scores. It is based on spatial domain NSS features derived from NSS model [32] and is not related to any specific distortion type.

In this thesis, subjective IQA and two types of objective IQA – Spatial-Spectral Entropy-based Quality (SSEQ) Index [11] and Natural Image Quality Evaluator (NIQE) [12] – are evaluated.

Chapter 3

Methodology

3.1 Experimental Overview & Nomenclature

In this section, an overview of the experimental setup is described briefly before moving on to the details explained in the following sections. The schematic diagram of the experiments conducted is given here (see Fig. 3.1). The main contributions of the thesis are data collection (image and video), data augmentation, developing a modified version of InceptionResNet-v2 (CNN-IR) and a train-time fusion network (CNN-F).

The Bag of Words, HOG with SVM and baseline CNN-s network are trained on the original data, while the deep networks CNN-IR and CNN-F are trained on augmented image data. All the images are normalized for training. The trained models are then tested on the video dataset, where the bounding boxes are manually given, which acts like a detector and then these detected images are fed into the trained model for output label. The schematic for these experiments are shown in Fig. 3.2. Please note that the videos are only for testing and are not used in training the models.

Nomenclature: Five deep learning neural networks are presented in this thesis. For Simplicity, each network is given the following abbreviations.

(i) CNN-s : Initial baseline CNN on grayscale images (CNN-s-r: CNN-s trained on RGB images)

(ii) Inception-v3-Ro : Inception-v3 retrained model trained on original data

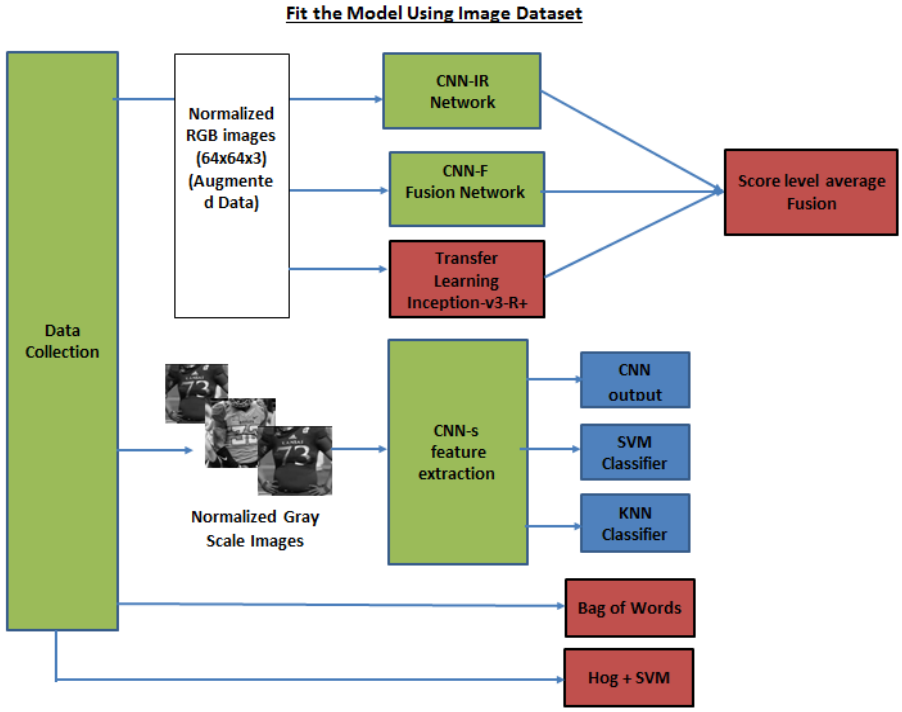

Figure 3.1: Overview of the experimental setup. The main contributions of the thesis (green) are data-collection of jerseys, trained CNN-s, CNN-IR and CNN-F on the dataset. Experiments are also conducted on methods like Bag of Words, HOG with SVM, transfer learning and score level average fusion (red). Classifiers used at the end of the networks are SVM, *k*NN and softmax (blue)

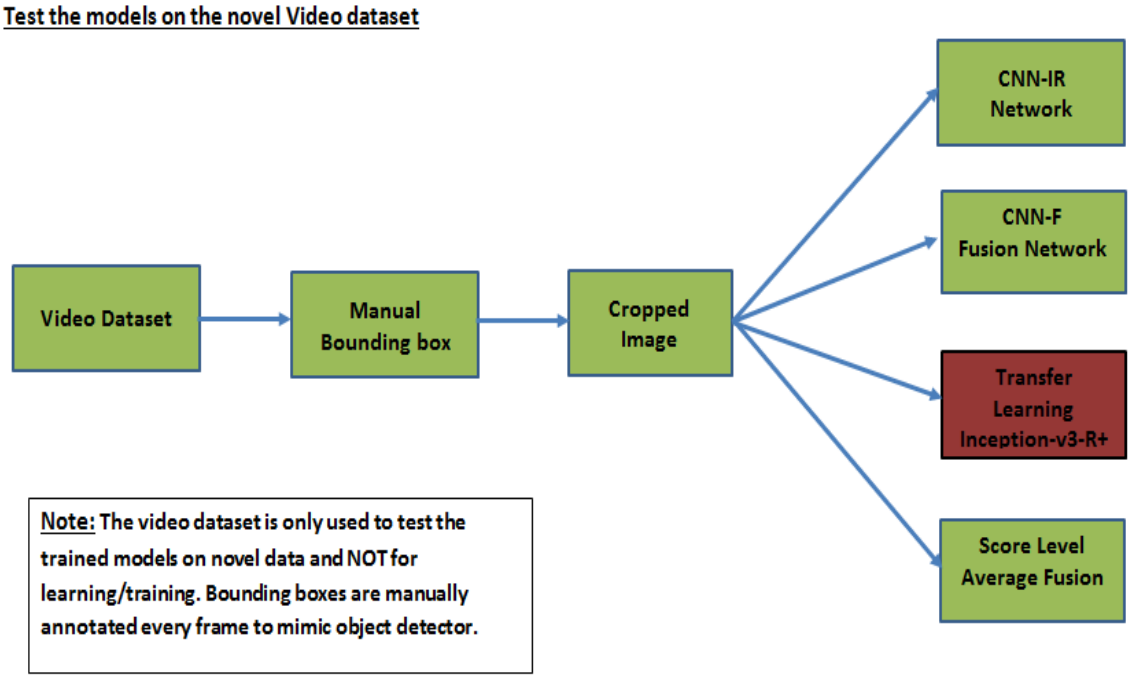

Figure 3.2: The experimental overview of tests conducted on video dataset.

(iii) Inception-v3-R+ : Inception-v3 retrained model trained on augmented data

(iv) CNN-IR : Modified InceptionResNet-v2

(v) CNN-F : Train-time fusion network

The abbreviations considered for the 14 teams in order shown in Figure 1.1 are as follows:

1. BYA - Baylor Bears Alternate
2. BYC - Baylor Bears Home
3. IOC - Iowa Cyclones Home
4. KJC - Kansas Jayhawks Home
5. KWC - Kansas Wildcats Home
6. OCB - Oklahoma Cowboys Alternate
7. OCC - Oklahoma Cowboys Home
8. OSC - Oklahoma Sooners Home
9. RRC - Texas Tech Red Raiders Home
10. TFA - Texas Horned Frogs Alternate
11. TFC - Texas Horned Frogs Home
12. TLC - Texas Longhorns Home
13. WVA - West Virginia Mountaineers Alternate
14. WVC - West Virginia Mountaineers Home

3.2 Dataset Collection

3.2.1 Image Dataset

As there are no publicly available jersey image datasets, a new dataset is generated. It was constructed by collecting frontal and off-angle jersey images from various sites and video sources when using different search engines. The average resolution of the image dataset is 325×326 pixels, with the least being 46×49 and highest being 2435×2441.

Cropping:

More than 10,000 images were downloaded and cropped such that the object of interest (jersey) is in the center using a Graphical User Interface (GUI), created to pre-process the collected original jersey images. The tool is used to load an image and, then, the GUI operator crops a square image by clicking the center of the image. A range of spatial resolution in multiples of 50:10:100 pixels, independent of the distance of the target to the source (camera) are cropped at once (see Fig 3.3). The best images including the jersey, regardless of bounding tightness are taken into the dataset by human observation. By using the pre-processing tool, all the images of subjects wearing jerseys are manually localized for further processing (i.e. to be used as an input to CNNs).

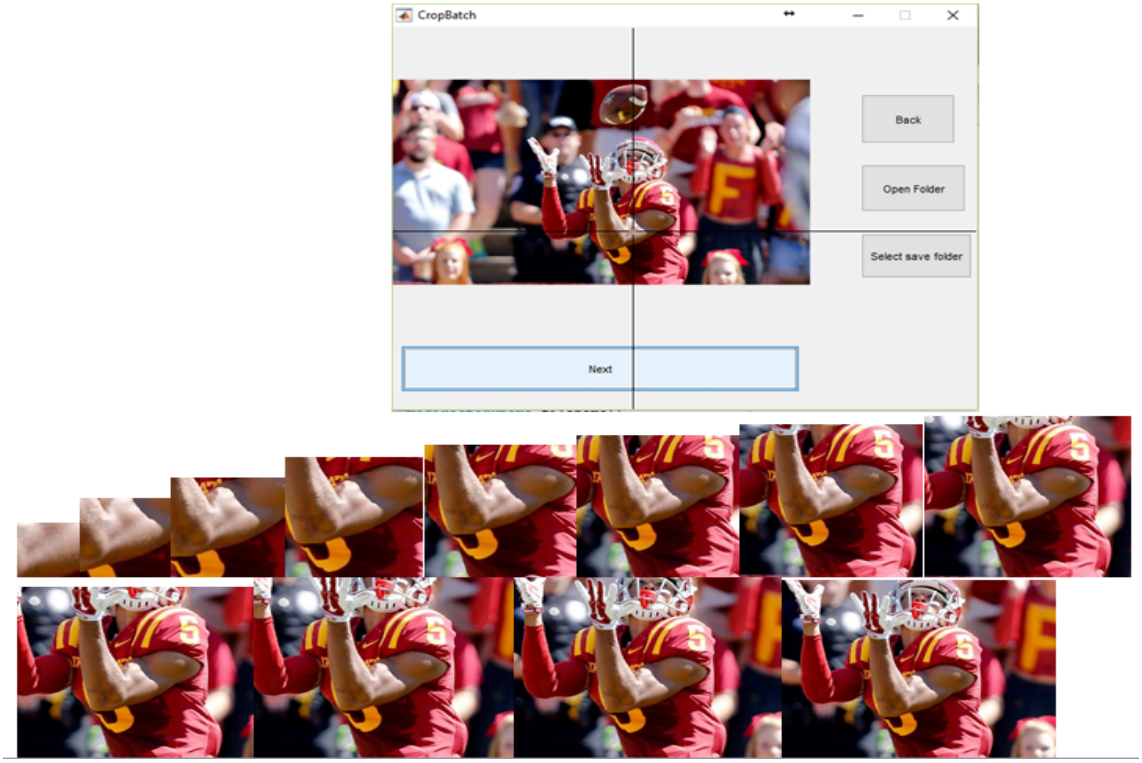

Figure 3.3: Cropping of the image by clicking the center of the assumed bounding box and the resulted output images, using MATLAB GUI.

A small set of jersey images of the JerseyXIV dataset are illustrated in (see Fig. 1.1). The final database consists of 14 categories with 500 train, 30 and 30 jersey images used for testing and validation respectively per class, i.e, originally, (before augmentation) a total of 7000 training, 420 test and 420 validation images.

3.2.2 Image Data Augmentation

The data is augmented by random horizontal shifting, rotating and random addition of Gaussian and salt & pepper noise. Matlab inbuilt functions are used for this purpose. The images are randomly rotated between [7°, 28°], horizontally shifted between x and y values of [15, 20] and randomly noised with Gaussian and salt & pepper noise at different random noise levels between [0.05, 0.3]. Sometimes, one image may also be combinedly shifted and added noise randomly. Based on the GPU resources available, the training and validation data are augmented 5 times, resulting in 35,000 training and 2,100 validation images. An example of the augmented dataset is provided in Fig. 3.4.

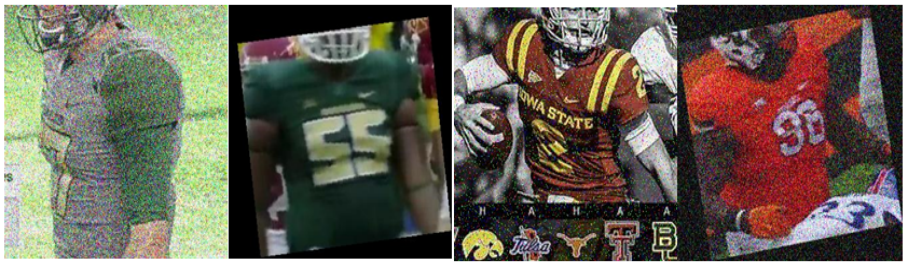

Figure 3.4: Example of augmented images - Gaussian noise, horizontal shift with rotation, salt & pepper noise, Gaussian noise on horizontally shifted and rotated image.

3.2.3 Video Dataset

The video dataset consists of 14 videos, one for each of the 14 classes. There are 3,584 total frames, with 2,188 containing the object. Detection of objects is implemented by specifying manual bounding boxes in each frame. MATLAB is used to annotate [x y h w] of the bounding box, where (x,y) is starting point, h and w are the height and width of the box. The values are calculated by choosing two diagonal corner points of the box intended to annotate. This method helped in reducing the time of annotating. An example of a frame with a bounding box can be seen in Fig. 3.5.

The following sections are briefly described as:

Part-1: Part-1 has the conventional methods - HOG with SVM and Bag of Words

Part-2: Part-2 consists of all convolutional neural networks- CNN-s, CNN-IR, CNN-F and transfer learning. Score level average fusion is also explained here

Figure 3.5: Example video dataset frames of Iowa Cyclone category. The green box is the provided bounding box.

Part-3: Part-3 has the application of the learned models on the video dataset and image quality assessment on the video dataset for improved performance.

3.3 Part-1: Conventional Classification Methods

Before going on to convolutional neural networks, two of the conventional methods were tested for the classification of JerseyXIV dataset.

3.3.1 Histogram of Oriented Gradient (HOG) with an SVM

Histogram of oriented gradient is a well known feature descriptor for object recognition. It is an edge oriented histogram based on the orientation of the gradient in localized regions [15]. The RGB train images are converted into gray scale and HOG feature vector is constructed using MATLAB built in function. A linear SVM classifier is modeled on the extracted HOG feature vector to classify into 14 classes.

Support Vector Machines (SVM) are a supervised learning algorithms used for analyzing the data for classification or regression. It was primarily a binary classifier and can be extended to multi-class classifier using 'one vs all' method. Given training data with ground labels, the algorithm builds a model with an optimum hyper planes/margins dividing the data, grouped according to their labels. The margin is a decision plane which defines the decision boundary.

The classification can be linear or non-linear based on the kernel type used - linear, polynomial, radial basis and sigmoid. LibSVM [33] provides a library for modeling these parameters and is used in this thesis for the same.

3.3.2 Bag of Visual Words (BOW)

Bag of Words works by creating a histogram of visual word occurrences representing the image and classification is performed by training an image category classifier on these histograms. In this thesis, Matlab inbuilt BOW function is used to test this method on the Jersey XIV dataset and the method follow the paper by Csukara et al.[16].

As the first step in the BOW method, feature descriptors are extracted from the input images. Here, SIFT is used to detect and extract the key point descriptors of interest points. Scale Invariant Feature Transform (SIFT) is a fairly robust detector and descriptor which is being used in many computer vision projects. It is invariant to image rotation, scale and robust across a substantial range of affine distortion, noise and change in illumination [34]. Next, a set of vocabularies are constructed using kmeans clustering. Each observation is assigned to a clusters whose squared euclidean distance is the least and the centers of these clusters are called a vocabulary. The number of key points nearest to each centroid in each class are represented as a histogram. This histogram represents the bag of keypoints which are used as a feature vector to train a multi-class SVM (Support Vector Machines) classifer.

3.4 Part-2: Convolutional Neural Networks

Convolutional Neural Network is a type of artificial neural network whose layers resemble the simple and complex cells in the primary visual cortex [17]. The connectivity is inspired from the biological brain neurons, where each neuron receives input signals (x_0) from its dendrites and produces output signals along its axon, which in turn connects to different other neurons. The signals are multiplied with the dendrites ($x_0.w_0$) of other neurons while passing along the axons. The signals are only sent (fired) through the axon when the summation is above a certain threshold, which can be modeled as activation function f [1]. This phenomenon

can be shown as a mathematical model as in Fig 3.6.

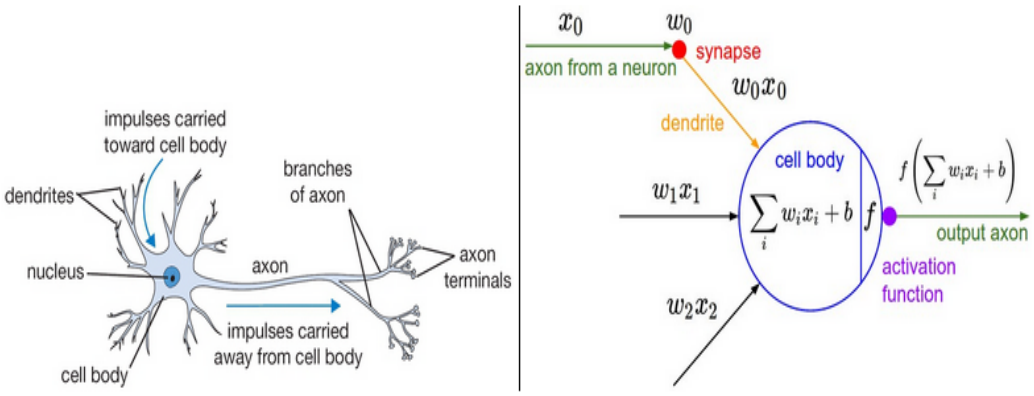

Figure 3.6: Biological Neuron and its mathematical model [1]

The CNNs have local connectivity property, where each neuron is only connected to a small region (receptive field) of adjacent layers. The neurons in the layers of a CNN are three dimensional (height, width and depth) and are connected only to the receptive field of the previous layer. CNNs have weight sharing property and form a feature map with replicated units sharing the same weight and bias, which makes all the neurons in the layer detect the same features. This feature decreases the parameters and so helps in reducing the memory cost.

The architectural overview of convolutional neural network is composed of distinct layers, which transforms the input to output through certain functions. The layers and activation functions are described below.

3.4.1 Architectural Components

(i) Convolutional layer:

Convolutional Layer is the core building block of the network. It consists of a set of learnable filters with small receptive fields but extends through the full depth of the input volume. When the input passes through the convolution layer, each filter is slid across the width and height of the input volume computing the dot product between the entries of the filter and the input at that position. This produces a two dimensional activation map and these activation maps are stacked along the depth dimension for all filters, producing the output of the convo-

lution layer.

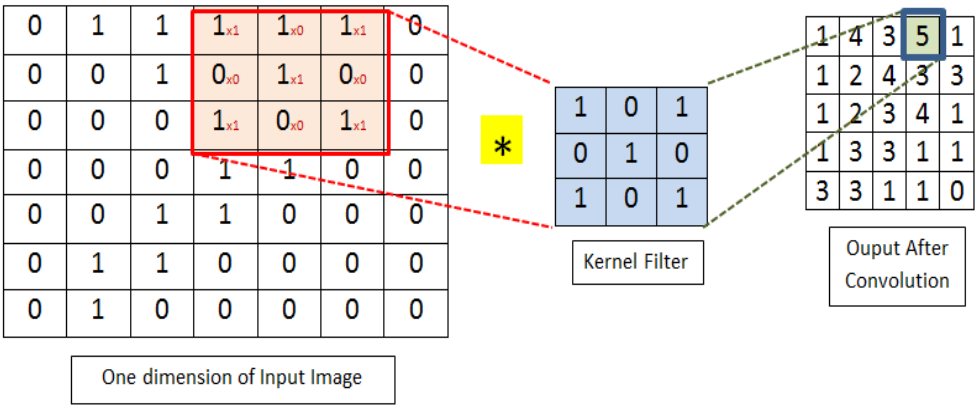

Figure 3.7: Illustration of convolutional layer output calculation

(ii) Max Pooling layer:

Pooling is a form of non-linear down-sampling method used to achieve spatial invariance by progressively reducing the spatial size of feature maps and the number of parameters. Pooling layer is also called as subsampling layer. There are many variants of pooling, such as average pooling, L2-norm pooling, etc., in which Max pooling is a popular one, outperforming the other subsampling operations [35].

In the construction of neural networks in this thesis, max pool layer is used as sub-sampling layer. Generally, pooling layer comes after convolutional layer, but is not necessary. The most common pooling layer window size is of 2×2, with a stride of 2 and is applied independently along the depth of the input, by which the width and height is reduced by 2 while the depth remains same. The max pooling function applies the window to the input patch and computes the maximum in the neighborhood as shown in Fig. 3.8.

(iii) Rectified Linear Unit layer:

This layer applies a non-saturating nonlinearity activation function to model the neuron's output [36]. There are different kinds of activation functions such as, tanh, sigmoid, softplus, etc. Rectified Linear Unit (ReLU) is used in this thesis, as ReLUs are said to train several times faster than their equivalents with tanh units and also reduce over-fitting [8].

The ReLU activation function is defined as

$$f(x) = max(0, x)$$

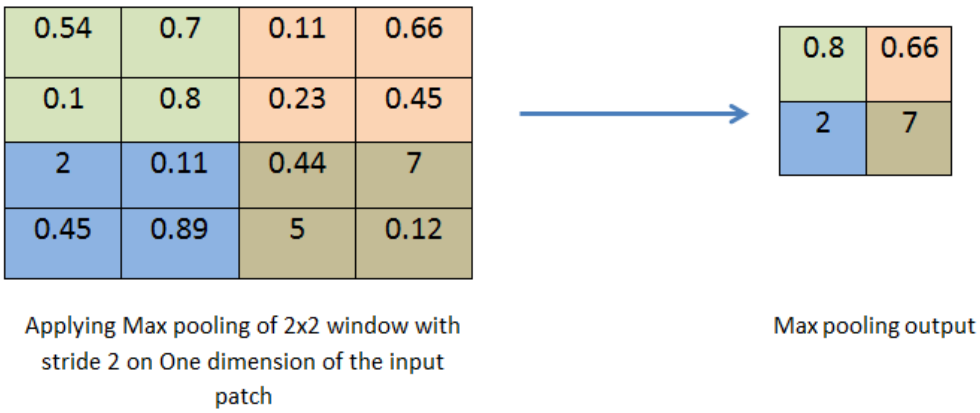

Figure 3.8: Illustration of Max pooling layer calculation

The plot of the Rectified Linear Unit function is shown in Fig. 3.10 . The function thresholds the activations at zero. This creates a disadvantage that if at all any neuron is updated to zero, then the gradient passing through it will always be zero since the gradient of a zero is a zero. Generally, ReLU is added after each pair of convolution and pooling layer.

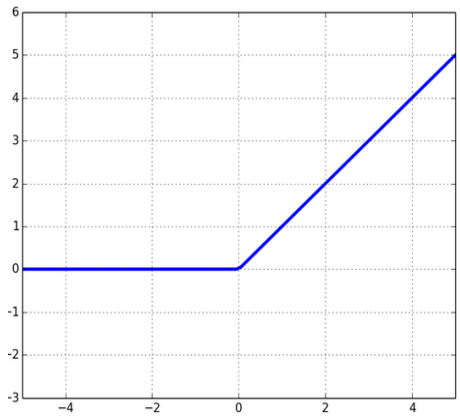

Figure 3.9: Plot of ReLU function

(iv) Fully Connected Layer:

In the fully connected layer, neurons have connections to all activations in the previous layer. It is generally used as the last layer which holds the output scores and is also called "output layer". Its activations are computed with a matrix multiplication and bias offset. The output of the fully connected layer is $1 \times 1 \times C$, where C is the number of class labels.

3.4.2 Employed Design Components

In this section, the design methods chosen for training the CNN-s, CNN-F and CNN-IR are explained. While the individual architectures are explained in their respective sections, the design steps taken for modeling the data and the network are explained as the following.

(i) Data Preprocessing:

Data preprocessing is a method used to transform the raw input data into an evenly distributed, scaled data within a range, understandable to the network. Data preprocessing is proven to improve the training process of the neural networks. Data can be preprocessed by mean subtraction or normalization. In this thesis, all the training data is normalized by centering the data to have a zero mean and scaling to [-1 1] range.

(ii) Weight initialization:

The network weights are initialized before training in order to avoid the diminishing of the input signal variance as it passes through the layers. In CNN-s, which is not a deep network, the weights are initialized randomly which are scaled by 1/sqrt(n). This scaling of inputs ensures that all the neurons in the network initially have approximately the same output distribution and empirically improves the rate of convergence [1]. Very low weights shrink the signal, while with very large weights the signal grows at each layer and becomes useless. In order to have the weights not too low or large, Xavier initialization [37] is introduced. Here, the weights are initialized from a distribution with zero mean and variance as in the formula:

$$Var(W) = \frac{2}{n_{in} + n_{out}}$$

where n_{in} and n_{out} are number of inputs and outputs of the layer. This weight initialization is applied for CNN-IR and CNN-F networks.

(iii) Batch Normalization:

Ioffee and Szegedy [38] address the internal covariate shift, a phenomenon named as the change in the distribution of network activations due to the change in parameters during training, which needs a careful parameter initialization and lower learning rates. They have introduced Batch Normalization method to avoid saturating nonlinearities by applying normal-

ization of the layer inputs for each training mini batch. Batch normalization is only used in CNN-IR and CNN-F networks and not in CNN-s.

(iv) Regularization Method: Drop out:

The neural network is prone to overfitting, a phenomenon where the network has a very low training loss but could not generalize to novel data. Dropout is a regularization method to address the overfitting issue to some extent and has shown to improve performance [2]. The dropout technique drops out the units in a neural network below a certain threshold p, where p is between [0, 1]. A dropout of 0.5 is used in CNN-s network and 0.8 in both CNN-IR and CNN-F networks.

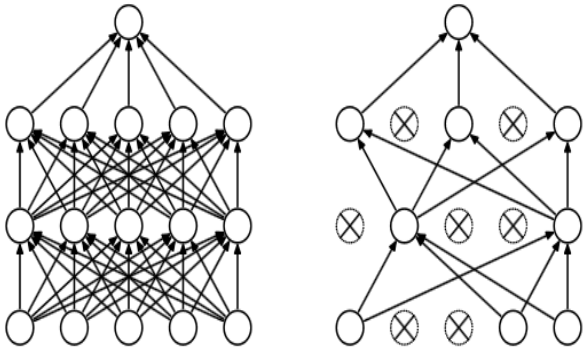

Figure 3.10: Schematic diagram showing the thinning of neural network after dropout [2]

(v) Softmax cross-entropy Loss:

Softmax layer is used as the classifier layer in the neural network and the network is trained under a loss function. In the networks in this thesis, cross-entropy loss is used as cost function to train the network. The cost function is used to calculate the loss between the predicted and true labels and this layer will be right after the fully connected layer. The softmax function scales the scores to be in [0,1] range and all scores sum to one.

(vi) Optimization Method: Stochastic Gradient Descent:

Optimization method is used to minimize the loss function. Gradient descent is an iterative method of minimizing a differentiable loss function J(θ) by updating the parameters in the negative direction of the gradient of the loss function. Stochastic gradient descent is a variant of gradient descent, which performs parameter update in the negative direction for each training iteration to minimize the gradient. The gradient is computed and the weights are updated over

small batches of training data for every iteration. The general form of stochastic gradient can be given as

$$\theta = \theta - \eta.\nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

where η is the learning rate or step size. The learning rate can be kept constant or changed for every or after certain iterations. There are other methods of gradient descent algorithms: Adagrad, Adam, RMSprop and Adadelta, which after empirical evaluation of these methods, stochastic gradient descent (SGD) is employed in all the trained neural networks in this thesis.

3.4.3 Baseline CNN-s Network

CNN-s is the abbreviation given to the following architecture and is considered as a baseline convolution neural network in this work.

3.4.3.1 Architecture

A simple CNN network is constructed and trained using the MATLAB Matconvnet software. This network is very simple with 11 layers and is considered as a baseline network in this work. It takes grayscale images for training and testing. The CNN-s network architecture built for JerseyXIV dataset consists of 11 layers, where each convolutional and maxpool layer pair are followed by a ReLu layer. The fully connected layer is followed by a softmax layer, which calculates the probabilities and the label with the highest probability will be assigned as the final output.

The detailed network architecture is presented in Table 3.1. Each input RGB jersey image, after resizing, is normalized to a 64×64 grayscale image. Each pair of Conv+Maxpool is followed by a rectified linear layer, which allows the network to train several times faster [8]. A dropout layer is added as a regularization layer to prevent overfitting [2]. The final layer, after the fully connected layer, is a softmax layer that classifies the input jersey image into one of the 14 classes.

Table 3.1: CNN-s architecture with grayscale image input size of 64×64. A rectified Linear Unit is applied after each Maxpool layer

Layer Type	Patch size/stride	Output size
Convolution	5×5 /1	60×60×30
Max pool	2×2 /2	30×30×30
Convolution	3×3 /1	28×28×80
Max pool	2×2 /2	14×14×80
Convolution	5×5 /1	10×10×500
Max pool	2×2 /2	5×5×500
Dropout	0.5%	5×5×500
Fully Connected	5×5 /1	1×1×14

3.4.3.2 CNN-s + K-NN Classifier

K-NN is a non-parametric algorithm used for classification. For a given test input, the algorithm computes a distance metric with the training data and according to the majority of k-nearest neighbors, the test input is classified [39].

In this method, the scores from the CNN is given as a feature vector to the KNN classifier and is modeled for 14 classes. A series of experiments are performed in determining the K value and distance metric, which are presented in the Experiments and Results chapter.

Score normalization:

Normalization is used to improve the speed by reducing the difference of the data [40]. The scores obtained from the CNN-s are preferably normalized to get best results. Two types of score normalization are evaluated– Soft normalization and Hard normalization, computed using the following formulae (3.1) (3.2) [41].

Soft Normalization:

$$\text{Norm} = (\text{X} - \text{Mean}) / (\text{StdDev}) \tag{3.1}$$

X — input score Vector

Mean — mean of the input vector

StdDev — Standard deviation of the input vector

Hard Normalization:

$$\text{Norm} = (X - \text{Min}) / (\text{Max} - \text{Min}) \tag{3.2}$$

X — input score Vector

Min — Minimum value of the vector

Max — Maximum value of the vector

3.4.3.3 CNN-s + SVM classifier

Linear Support Vector Machines (SVM) are widely used for binary classification and are based on statistical learning theories [42]. SVM is relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space [43], which makes this method learn a larger set of patterns.

In this work, LibSVM [33] library is used for Multi class SVM training. One vs. All method with an RBF Kernel is applied, where a separate binary model for each class, as positive and all other as negative, is trained. Finally, the highest probability of all the models, for a given input, is outputted as the predicted label. The scores are again normalized as in (3.1) (3.2) formulae for the hard and soft score normalization.

3.4.3.4 CNN-s-r Network

The baseline CNN-s network is trained on grayscale input images and on the original (not augmented) data. To make a fair comparison with the convolutional neural networks in the upcoming subsections, the CNN-s architecture is trained on RGB input images on augmented data using Tensorflow library.

The architecture of CNN-s is not modified except that the input takes a $64 \times 64 \times 3$ and that it is trained using tf-slim library instead of Matconvnet library. CNN-s-r is the abbreviation given to the baseline architecture with RGB input images.

3.4.4 Inception-v3 Transfer Learning

The recent classification models are being developed with high depth [3] [9] and trained on a high amount of data [44] [8]. So, the models have millions of parameters and can take few weeks to train. Inception-v3 [9] is one of those deep neural network models which is trained on an ILSVRC-2012 dataset containing 1,000 classes. Also, a sufficient dataset size is required for the depth of the designed network for better training [45].

Transfer learning is used to train a model on small or of insufficient sized dataset to achieve good performance results, where a network trained on a larger dataset is used to initialize the weights. Transfer learning is an approach applied for leveraging the knowledge of already learned networks, whose extracted features are found to be very powerful in object detection and classification problems [46].

The Inception-v3 architecture is based on GoogLeNet [47], which aimed at decreasing the computational parameters. The focus of building Inception-v3 was put on to find a tradeoff between general design principles, such as size, depth and sparsity in the layers, to increase the performance as well as to decrease the computation cost. The network achieved 21.2% top-1 and 5.6% top-5 error for ILSVRC-2012 dataset. It has also reported using much less computational power than denser networks [47] and relatively 2.5$\times$ increase when compared to the network described by [38]. Please refer [9] for inception-v3 full architecture details. In this thesis, the last layer of trained inception-v3 network is retrained on JerseyXIV dataset.

3.4.5 CNN- IR

CNN-IR is the abbreviation for the network modified from Inception-ResNet-v2 [3]. Inception-ResNet-v2-network is trained on ILSVRC 2012 dataset with 19.9% top-1 error. It combines the optimization benefits of residual connections [20] with the computational efficiency of Inception units [48].

3.4.5.1 Choice of Network Depth and Input Image size

After analyzing the original network, the network depth as well as the input image size is changed for the following reasons:

(i) Change in network depth: The original network was trained with the augmented JerseyXIV dataset with no observation of learning, as the loss curve did not drop down even after many epochs. The training loss value is 1.178 for 3,500 steps (see Fig. 3.11), whereas the CNN-IR network has the loss value of about 0.45 at 3,500 steps (see Fig. 4.5). This phenomenon is established as that the original network's depth is large for the dataset size to have any significant features reach the bottom layers for learning without getting diminished in midway [45]. So as to utilize the Inception-ResNet-v2's both residual and inception block's proven learning capacity and also to accommodate the network to the size of the dataset, the network depth is modified.

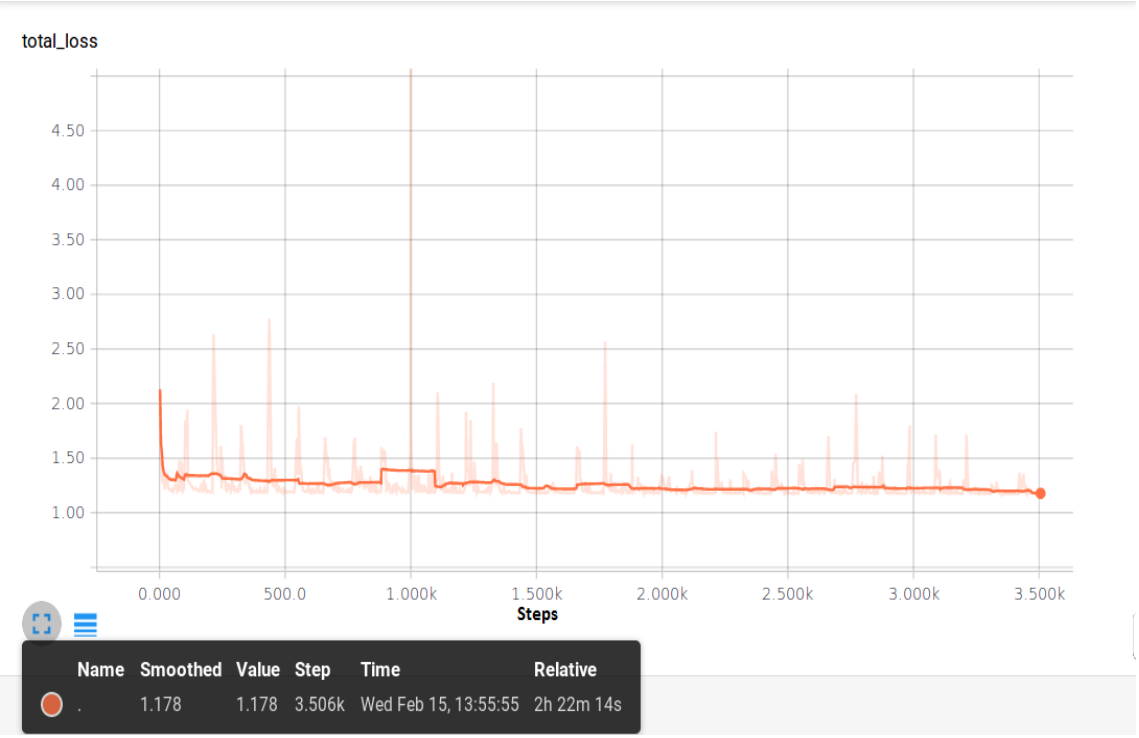

Figure 3.11: Training loss curve of InceptionResNet-v2 on Jersey XIV dataset.

(ii) Change in input size: Most of the images of the dataset are collected from videos and after localizing the image by cropping, the average resolution is 325×326 pixels, with the least being 46×49 and highest being 2435×2441. The ILSVRC-2014 datasets have an average image resolution of 482×415 pixels [44] (The average resolution of ILSVRC 2012 on which the InceptionResNet-v2 was trained, was not provided). The size of the input image can be related to the depth of the network and computational cost. The higher the image size, the higher the features and cost. In addition, experiments for training CNN-IR, with input size of

299×299 is conducted with a batch size of 32 and the results can be seen in Figure 3.12. It can be seen that the training for 20k steps have taken more than two days while 64×64 input size with batch size of 64 took one day for 50k steps (see Fig. 4.5). To minimize the loss of information by enlarging smaller images and to have a significant increase in training time, a trade off can be made by decreasing the input image size.

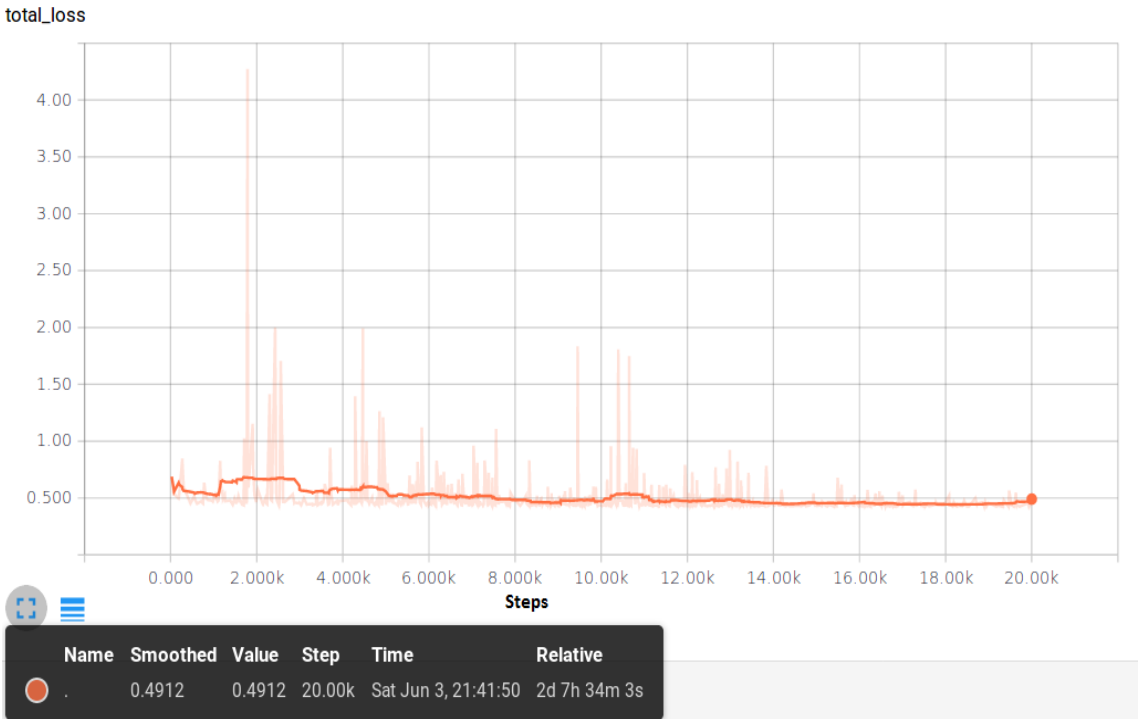

Figure 3.12: Training loss curve of CNN-IR with input size of 299×299 .

Please refer [3] for full architectural details of the Inception-ResNet-v2 network. The following section will only discuss the modified network and blocks which were taken from the original network.

3.4.5.2 Architecture

The architecture of CNN-IR has one module of inception network and 10x Residual network module, in contrast with 40x Residual modules and 3 inception network blocks in the original network. Please refer Appendix (see Fig. 2) for viewing a schematic diagram of Inception-ResNet-v2 original network.

In contrast with the original network with input 299x299x3, the input of the CNN-IR is a 64x64x3. The image size is reduced as the average resolution of the JerseyXIV dataset is not

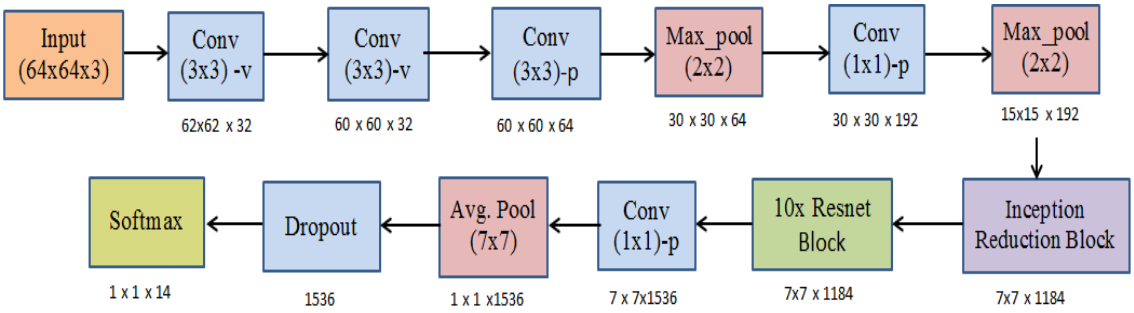

Figure 3.13: CNN-IR Architecture based on Inception-ResNet-v2.

as high as ILSVRC dataset and so to avoid losing the features by enlarging. The architecture of CNN-IR is shown in Fig. 3.13. All convolution layers that are marked with P are same-padded, i.e, their output grid size matches with their input. The convolution layers which are marked V are valid-padded, i.e., the grid size of output is reduced accordingly by the formula:

Output Width = (W+2P-F) /S + 1; W= width

Output Height = (H+2P-F) /S + 1; H= height

F- filter size, P = pad, S = stride

The layer parameters, filter size, pooling size, etc., are changed from the original network, according to the above formula, so that the end softmax layer has a 1x1x14 output size, representing 14 probabilities for each of 14 classes. The residual module and inception module are shown in the following figures (see Figs. 3.14 3.15).

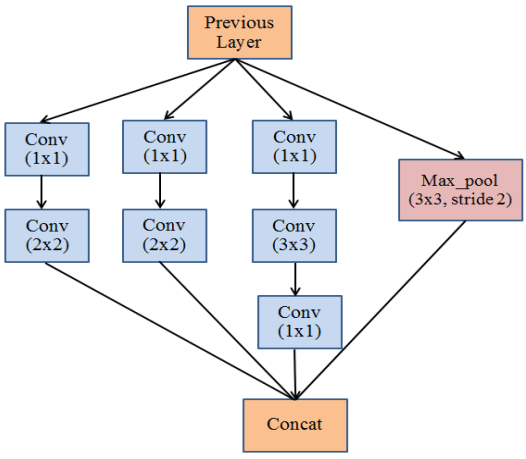

Figure 3.14: CNN-IR 15×15 to 7×7 reduction block [3].

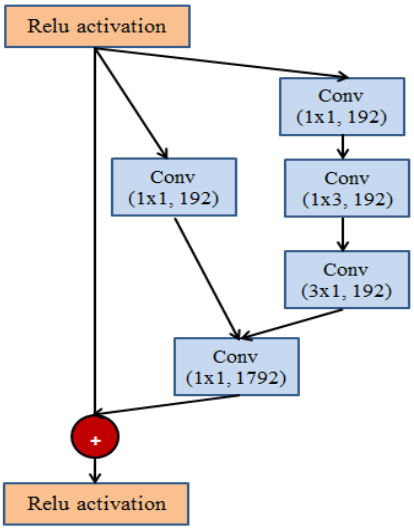

Figure 3.15: CNN-IR ResNet Block [3].

3.4.6 Proposed CNN-F Network

CNN-F is the proposed network for the classification of the Jersey XIV image dataset. After seeing the network in network technique of the inception architectures, an idea to combine two networks at the training phase was implemented. Many attempts were made in order to determine at what layer or after how many layers the features has to be combined in a vector. Here, the network which has achieved the highest test accuracy of 92.61% is discussed. The fusion done here is feature level.

3.4.6.1 Architecture

CNN-F is a fusion of two networks, of which one is CNN-IR and the other CNN network consisting of 7 layers with each conv layer is followed by maxpool and ReLu activation function. Both the networks take in the image input and the features extracted from them are combined at feature level and the loss is calculated accordingly. The schematic diagram of CNN-F architecture is shown in figure 3.16. Both the networks take the same batch of input images. Following [3], only top layers are batch normalized [38] and the weights are initialized using Xavier initialization. Unlike the first network in CNN-F, where only small filters (3×3, 2×2) and deep layers are used, the second network uses higher filter sizes and significantly shallow layers in an attempt to obtain different features from both networks and increase the image information.

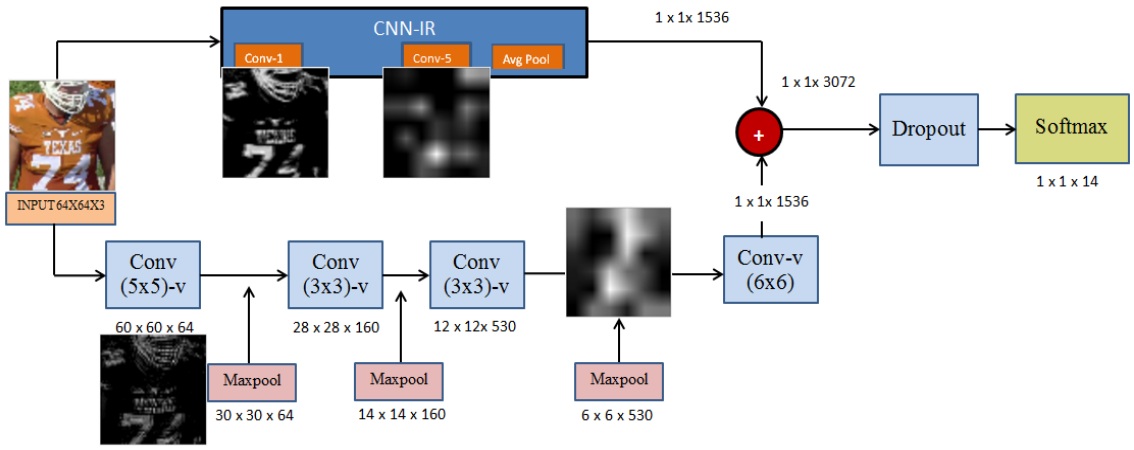

Figure 3.16: CNN-F: Fusion of two networks

3.4.7 Score Level Fusion

Fusion is a method of combining the scores from different classifiers in an attempt to improve the classification performance. Even though there are many types of classifier fusion schemes [49], the most commonly used are the following - feature level fusion and score level fusion.

Feature level fusion: In this method, the features extracted from the trained models are fused into a vector.

Score level fusion: In this method, the features are extracted from different classifiers and their individual soft output scores (between [0,1]) are fused by minimum, maximum, product and average fusion rules. These techniques are class-conscious fusion techniques for soft labels [50].

In this thesis, score level fusion with average rule is used. The fusion of classifiers does not produce any better results if all the classifiers produce same errors or are correlated. After doing an empirical study on the different fusion methods (results provided in Appendix 5.2), it is established that the average score level fusion of Inception-v3-R+, CNN-IR and CNN-F have shown improved accuracy.

3.4.7.1 Score level Average Fusion

Following [50], simple aggregation rule for average is used.

Let for an test input and Class labels =(1, 2,.., C), the scores for each classifier be

s1 =$\{ \alpha_1, \alpha_2 ....., \alpha_C \}, where s1 \in [0, 1]$

s2 =$\{ \beta_1, \beta_2 ....., \beta_C \}, where s2 \in [0, 1]$

s3 =$\{ \gamma_1, \gamma_2 ....., \gamma_C \}, where s3 \in [0, 1]$

Then the average fusion of the score be,

S[i] = (s1[i] + s2[i] + s3[i]) / 3 ; for i $\in C$

The output label is the max(S).

3.5 Part-3: Application in Videos

To test the novelty and application of the trained models, a video test data is created. The video dataset contains 3,584 total frames, with 2,188 containing the object. The frames are manually annotated with a bounding box and is fed into the neural network. The schematic diagram in (see Fig. 3.17) shows the overall experimental process.

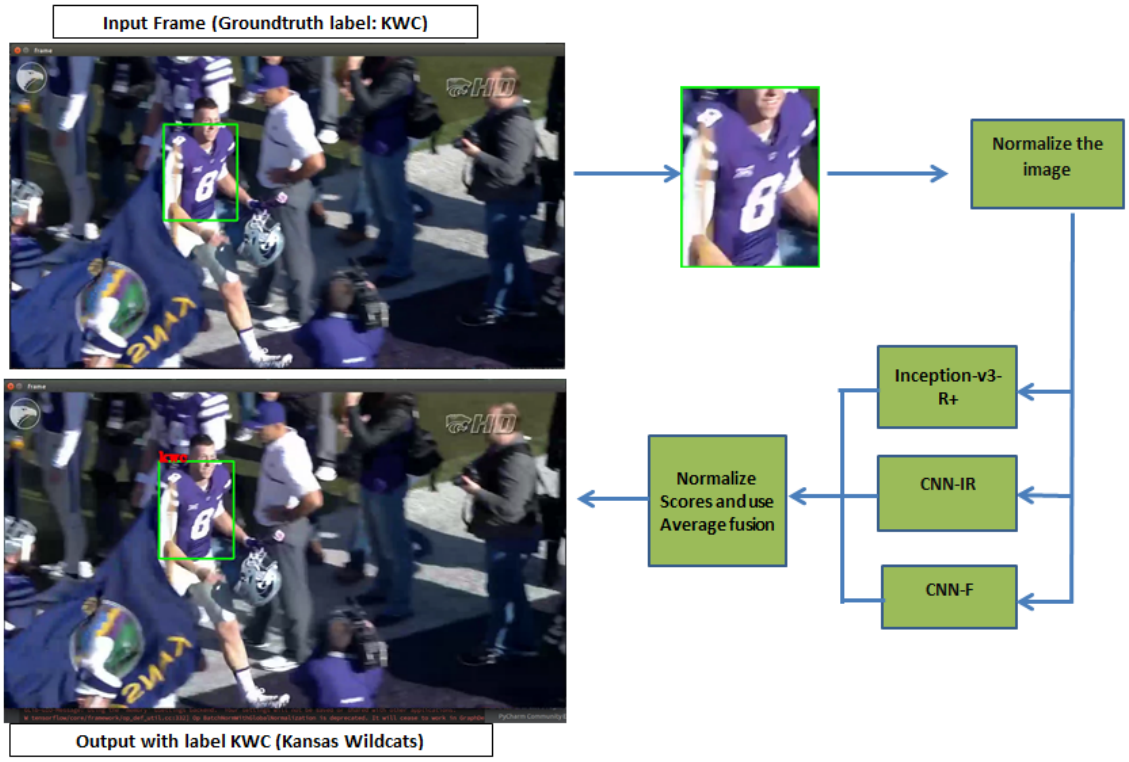

Figure 3.17: Schematic diagram showing the experimental process for average fusion of scores. Other methods are also tested the same way individually, without the average fusion.

3.5.1 No-Reference Image Quality Assessment

Football jersey classification in a streaming football game has challenges other than the game itself (occlusions, illumination, pose, etc.). The videos are subject to distortions like noise, blur and compression during capturing, transmission, processing and reproduction. These distortions can seriously affect the information available for classification purposes.

The Average-fusion model has achieved 81.31% accuracy on the video dataset. It has correctly classified 1,779 images and misclassified 409 images out of 2,188 input test images from the video dataset. In order to answer the question if the distortions in the video test dataset (bounded image per frame) affect the performance of the Average-fusion model, Image Quality Assessment (IQA) is employed.

As discussed in the literature review section, a no-reference or blind QA is employed as there is no available reference image. In this case where the classification is done in real-time, reference image does not exist and this leaves the only option of no-reference or blind IQA. The no-reference subjective IQA and objective IQA conducted on the images (bounded image per frame) which are classified and misclassified from the Average fusion model are explained.

3.5.1.1 Subjective Image Quality Assessment

Before conducting the tests for NR-IQA, an initial subjective analysis of the data being fed into the neural network for classification would help in understanding the distortions in the images from human perception. Certain criteria are considered in classifying the images accordingly. The criteria based on how the images are classified by a person are shown in Fig. 3.18. Five criteria for subjective analysis are considered and explained as follows:

Blur/Low quality: The very visible distortion for the naked eye is blurriness. When the player runs across the field and if the shutter speed is lower, blurriness occurs. This distortion has a high rate of occurrence and is very obvious in this video dataset.

Occlusion: Occlusions can reduce the required information in the image. Here, any image containing an object (hand, ball or other player) occluding the jersey is considered as an occluded image.

Side pose: The player pose can reduce the information, as the front of the jersey holds characters, symbols, etc. So any side angled pose falls into this category.

Good: Images with frontal pose or slightly side pose with all the information contained and free of visible blur fall into this category.

All three: There are a few images which are side posed, blurred and occluded images. These images fall into this category.

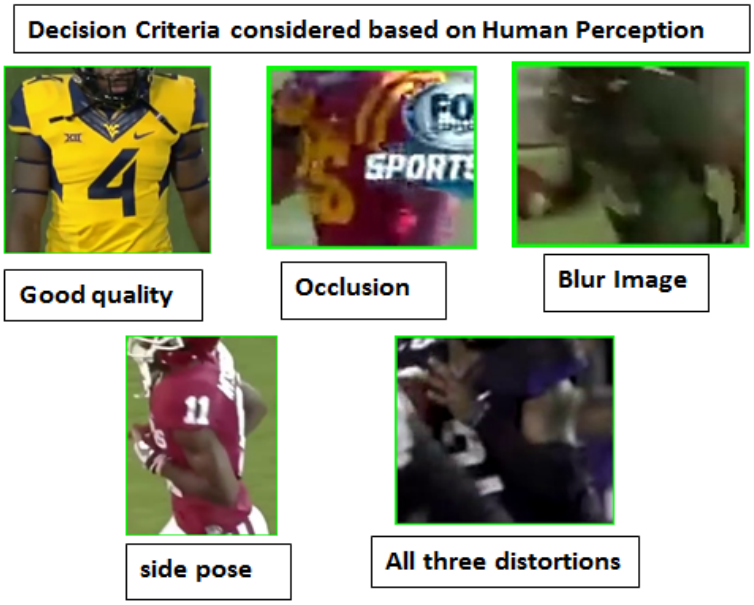

Figure 3.18: Decision criteria for Subjective IQA. Images are taken from classified and misclassified images by average-fusion method on video dataset.

3.5.1.2 Objective Image Quality Assessment

The subjective IQA method performed in the previous section may not be accurate and will be too slow to incorporate into real-time. Objective IQA methods such as reduced-reference or no-reference are design oriented methods where a system is modeled to predict the image quality automatically and accurately. No-reference IQA is the method used when the reference image is not available, such as in this case.

In this thesis, image quality assessment is conducted using two publicly available no-reference IQA analyzing softwares: Spatial-Spectral Entropy-based Quality (SSEQ) Index [11] and Natural Image Quality Evaluator (NIQE) [12].

(i) Spatial-Spectral Entropy-based Quality (SSEQ) Index:

Spatial-Spectral Entropy-based Quality (SSEQ) Index is proposed by Liu and Bovik [11] which is capable of assessing the quality of the distorted image across multiple distortions such as : Gaussian blur, white noise, JPEG compression, JP2K compression and fast fading. The paper claims that the method is statistically superior to the full-reference IQA algorithms such as SSIM (Structural Similarity Index) [51] and other no-reference IQA methods like BIQI (Blind Image Quality Index) [28], DIIVINE (Distortion Identification-based Image Verity and INtegrity Evaluation index) [29] and BLIINDS-II [30].

SSEQ model utilizes local spatial and spectral entropy features of the distorted images. The input images are first preprocessed by downsampling and then the spatial and frequency entropies of local patches are calculated as features. These extracted features are sorted and pooled together. They followed two-stage framework [28] for no-reference image quality assessment to get the final image quality score. An SVM classifier is trained to compute the probability of occurrence of each distortion in an image and the regression functions are trained on each distortion type against human scores [11]. The final predicted quality score is calculated from the dot product of distortion probability vector and distortion-specific quality vector.

(ii) Natural Image Quality Evaluator (NIQE):

SSEQ model is trained on *a priori* distorted images associated with human opinion scores, making it an opinion-aware model like DIIVIVE, BLIINDS and BRISQUE. Unlike these opinion-aware models, NIQE is a first of a kind NSS (Natural Scene Statistic)-driven blind opinion-unaware model which does not require exposure to distorted images nor any training on human opinion scores and performs better than the peak signal-to-noise-ratio (PSNR) and structural similarity (SSIM) index [12]. This opinion-unaware and distortion-unaware IQA model is constructed by fitting a collection of spatial domain NSS features, derived from NSS model [32], to a multivariate Gaussian (MVG) model. The NIQE Index is not related to any specific distortion type.

3.6 Hardware and Software Utilized

- The Bag of words and HOG + SVM algorithms were trained and tested on NVIDIA Quadro k620 2GB GPU using MATLAB R2015b software [52]. MATLAB inbuilt functions were used in the code.

- The CNN-s network was trained and tested on NVIDIA Quadro k620 2GB GPU and NVIDIA TITAN X 12GB GPU using MATLAB R2015b and MatConvNet [53] toolbox for computer vision applications.

- The Inception-v3 transfer learning is done using Tensorflow's tutorial. CNN-s-r, CNN-IR and CNN-F were trained and tested on two NVIDIA TITAN X 12GB GPUs, using Tensorflow-slim, which is a Tensorflow library [4]and utilizing its distributed machine learning system.

In this section, datasets, the methodology of the conventional methods evaluated, architectures of CNNs used and the quality assessment methods are discussed. In the next section, the experimental results of all these methods will be presented.

Chapter 4

Experiments and Results

4.1 Part-1: Conventional Classification Methods

4.1.1 Histogram of Oriented Gradient (HOG) with an SVM

The RGB train images are converted into grayscale and HOG feature vector is constructed using MATLAB built in function. A linear SVM classifier is modeled on the extracted HOG feature vector to classify into 14 classes. The figure 4.1 shows the extracted HOG features of the input image. The Test accuracy is 49.048%. The experiments were repeated 4 times.

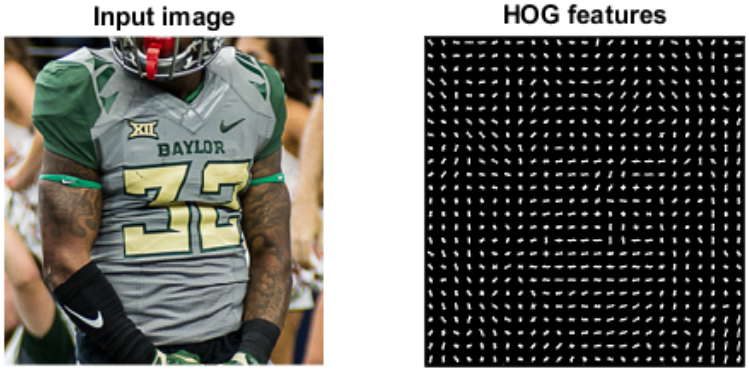

Figure 4.1: Extracted HOG features from the input image.

4.1.2 Bag of Words

As discussed in the methodology section 3.3.2, SIFT descriptor is used to extract the features. Bag of words, being a well established algorithm and freely available, MATLAB pre-

built code is used to train and test. The test results are presented in Table 4.1. The experiment for training and testing BOW is run 4 times. The table shows the accuracy of training set (giving train images for testing the fit of the model) and test image accuracy of each of the four experiments. The average of the training and testing accuracies four experiments are given in the final row, in bold.

Table 4.1: Results of BOW

Training Accuracy (%)	Test Accuracy(%)
70	53
71	55
71	59
70	57
Average Training Acc: 70.5	**Average Test Acc:** 56

4.2 Part-2 : Convolutional Neural Networks

4.2.1 Baseline CNN-s: Training and Results

CNN-s is the abbreviation given to the following architecture and is considered as a baseline Convolution neural network in this work. The following explains the training of the network and its results. The architecture is explained in section 3.4.3.1.

Training & Results

The CNN-s is trained for roughly 7500 epochs on normalized grayscale images, with a learning rate fixed at 0.001 throughout the training with a batch-size of 50.

The training of the network is limited by the memory available on the GPU and the amount of training time willing to be tolerated [8]. The training is run for 4 weeks on a 2GB GPU and allowed the network to progress until a pattern of gradual increase in validation error is observed, as shown in Fig. 4.2. The first layer features of the network can be seen in Fig. 4.3. The softmax layer is used as the classifier at the end of the network. It is a logistic function that computes the probabilities. The class with the highest probability is selected as the predicted label.

A top-1 validation error rate of 20% and top-5 error rate of 3.8% are finally achieved. Table

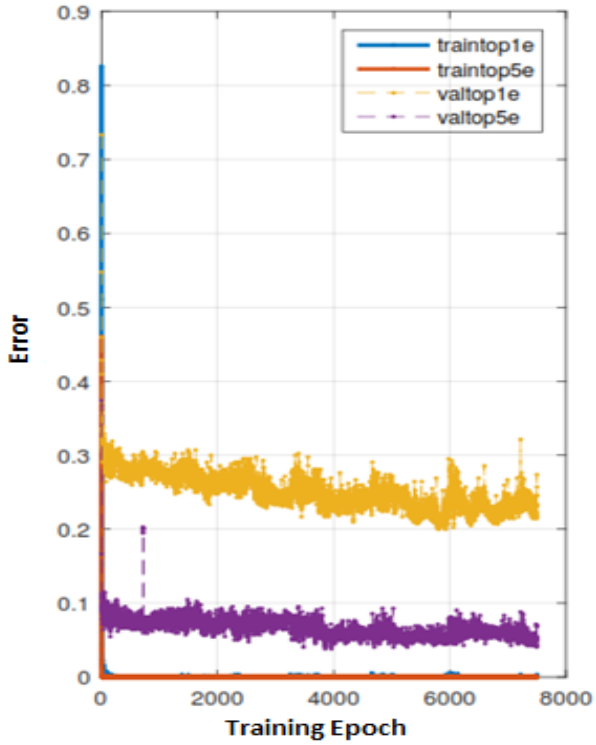

Figure 4.2: Training Epoch vs. Validation Error. Train top-1e and train top-5e have reached zero. A gradual decrease in validation error can be seen from val top-1e with increasing epochs and from 6000 epochs, a gradual increase in error can be seen which indicates overfitting.

4.2 shows the most satisfactory classification accuracy achieved in the test set at a particular epoch. It can be seen that the accuracy increases as the network is trained for longer epochs.

Table 4.2: CNN-s Training Results

No	Epoch	Test Accuracy (%)
1	757	77.14
2	1161	78.09
3	2537	80.24
4	4584	82.86
5	5483	83.00
6	6545	84.28

4.2.1.1 CNN-s + K-NN: Results

Parameter Selection:

An empirical study of K value and distance metric is conducted to determine the best K value for modeling the KNN classifier. Two types of distance metrics, namely euclidean and minkowski, were found to produce better results and different values of K are tested. These

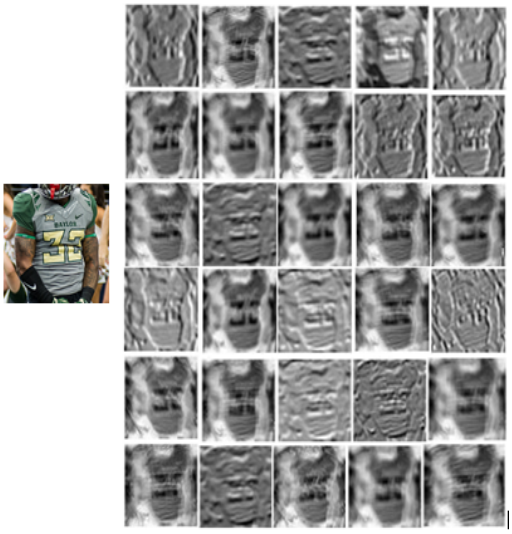

Figure 4.3: First Layer features of the input grayscale image from CNN-s network.

empirical evaluations are presented in the Table 4.3.

Table 4.3: Empirical study results for determining the K value and distance metric for modeling KNN classifier for raw, hard and soft normalized scores

No	K value	Distance metric	Normalization Method	CNN-s + KNN Test Accuracy (%)
1	16	Euclidean	Raw	82.85
2	16	Euclidean	Hard	80.95
3	16	Euclidean	Soft	81.90
4	32	Euclidean	Raw	81.90
5	**16**	**Minkowski**	**Raw**	**83.33**
6	4	Minkowski	Raw	83.09
7	3	Minkowski	Raw	82.61
8	16	Minkowski	Hard	82.14
9	**18**	**Minkowski**	**Hard**	**82.38**
10	20	Minkowski	Hard	81.48
11	18	Minkowski	Soft	81.66
12	16	Minkowski	Soft	82.38
13	**15**	**Minkowski**	**Soft**	**82.33**

Results:

Using the metrics determined from the empirical study for parameter selection, which are highlighted in bold in Table 4.3, each experiment is repeated 4 times. The best and average test

accuracies for each of raw, soft and hard normalized scores are presented in the Table 4.4.

Table 4.4: Results of CNN-s + *k*NN

K Value	Distance metric	Normalization Method	Test Accuracy (%)	Average Accuracy (%)
16	Minkowski	Raw	82.14	82.32
			81.20	
			83.33	
			82.62	
18	Minkowski	Hard	**82.38**	81.43
			80.48	
			80.95	
			81.90	
15	Minkowski	Soft	81.43	81.96
			82.14	
			81.90	
			82.38	

4.2.1.2 CNN-s + SVM: Training and Results

The scores of training data obtained from the CNN-s are again normalized as in (3.1) (3.2) formulae for the hard and soft score normalization.

Parameter Selection:

The hyper parameter selection is done by grid search using n-Fold cross validation. The grid search is done through an automatic selection of range of C and g values. This search range is changed automatically until best cross validation accuracy is found.

Through this study, the RBF kernel with cost and gamma parameters for raw, soft normalized and hard normalized scores respectively, C = 4.0629, 2.5491, 2.0629 and g =0. 015, 0.16494, 0.2192, are found to be optimum.

Results

After the parameter selection, One vs All SVM classifier is trained on the normalized scores of training data and tested on the test data, using the LibSVM library in MATLAB. The results are given in the Table 4.5. The table shows the results of tests repeated four times, the average and best of the four experiments.

Table 4.5: Results of CNN-s + SVM classifier

Cost (C)	Gamma (g)	CNN-s Test Accuracy (%)	CNN-s + SVM Test Accuracy (%)
Raw Score input			
4.0629	0.0015	80.24	83.57
		81.19	84.05
		80.71	83.81
		81.43	85.00
Average Test Accuracy		**80.89**	**84.10**
Soft normalized Score input			
2.5491	0.16494	81.43	85.24
		81.43	84.05
		81.19	83.62
		81.19	82.62
Average Test Accuracy		**81.25**	**83.88**
Hard normalized Score input			
2.0629	0.2192	81.43	83.29
		81.19	83.81
		80.24	82.86
		81.19	83.81
Average Test Accuracy		**81.01**	**83.69**

4.2.1.3 CNN-s-r: Training and Results

CNN-s-r is the abbreviation given to baseline convolutional neural network (CNN-s) with RGB input, while the CNN-s is trained on grayscale images. The CNN-s with RGB input is tested to make a fair comparison. The following explains the training of this network and its corresponding results. The architecture of CNN-s is explained in the methodology section 3.4.3.1 and the only change in CNN-s-r is that the input size is $64 \times 64 \times 3$.

Training & Results

The CNN-s-r is trained on Tensorflow tf-slim library for 120k steps on normalized RGB images, with a learning rate fixed at 0.001 throughout the training with a batch-size of 64. Softmax cross entropy loss and Stochastic gradient descent optimizer is used to train this network. These are the same parameters used to train CNN-s on grayscale images. The figure 4.4 shows

the training loss curve of the network. The loss value at 120k steps is 0.36 and 0.40 at 50k steps.

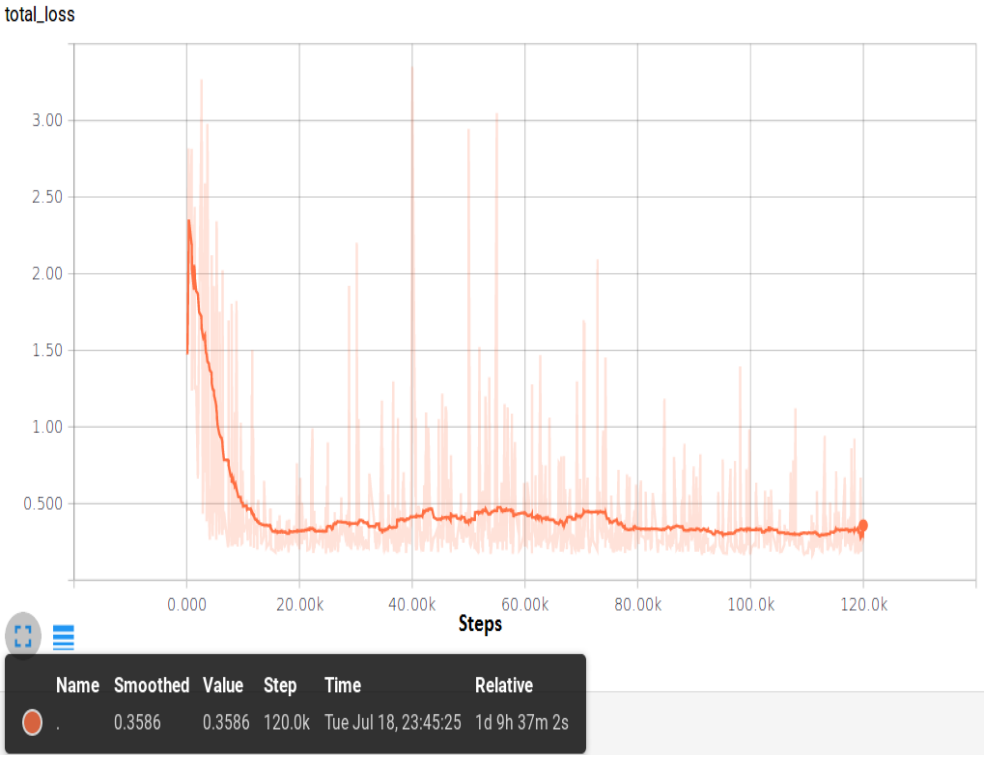

Figure 4.4: Training loss curve of CNN-s-r network.

The table 4.6 gives the test accuracy of the CNN-s-r model at certain steps through the training process on the image dataset. When compared to the training epochs of the CNN-s network, this network is trained for less epochs. The test accuracy is higher for CNN-s-r in lesser epochs than the CNN-s, i.e, CNN-s at 6545 epochs (916k steps, as 1epoch =140 steps for 7,000 training images with a batch size of 50) gave 84.28% test accuracy while CNN-s-r at 120k steps gave a test accuracy of 86.67%

Table 4.6: CNN-s-r Training Results

No	Steps	Test Accuracy (%)
1	10k	35.48
2	30k	41.90
3	50k	48.81
4	70k	59.76
5	90k	67.62
6	120k	86.67

4.2.2 Inception-v3 Retraining Results

Here, in this work, the Inception-v3 pre-trained network model is used as a feature extractor. The available Inception-v3 trained model on ILSVRC-2012 dataset is used with the tutorial code from Tensorflow [4] to retrain the network's final softmax layer for 14 classes of JerseyXIV.

Keeping all the layer weights of the model, only the final softmax layer is retrained for 14 classes. In short, the layers acted as a feature extractor and the final layer is trained on these extracted features to produce probabilities pertaining to 14 classes. The following describes the retraining of Inception-v3 on both original data and augmented data and their results.

(a) **Original Data (Inception-v3-Ro):** The network is trained on a raw JerseyXIV dataset containing 7000 training, 420 test and 420 validation images. The last layer of the model architecture is removed and retrained for 14 classes. 0.001 learning rate is used and is run for 100k steps. This method resulted in 88.27% test accuracy.

(b) **Augmented Data (Inception-v3-R+):** The same network model is retrained for 14 classes of the JerseyXIV dataset on the augmented by random horizontal shifting, rotating and adding random noise of Gaussian and salt & pepper. Based on the GPU resources available, the training and validation data are augmented 5 times, resulting in 35,000 training and 2,100 validation images. The training process was run for 100k steps with a learning rate of 0.001 and resulted in 92.38%.

4.2.3 CNN-IR: Training and Results

The CNN-IR network, as explained in the section 3.4.5.2, is trained on augmented data for 14 classes of jersey dataset with stochastic gradient descent on two Titan X GPUs, utilizing Tensorflow [4] distributed machine learning system and Tensorflow-slim library. A number of experiments were conducted and the best parameters for training are found to be 0.04 learning rate, which is reduced by half whenever a saturated loss curve is observed (see Table 4.7). A dropout rate of 0.8 is used. The network is trained for 50k steps with a batch size of 64, i.e., 86 epochs. Softmax cross entropy is used as the loss function. A test accuracy of 91.42% is achieved after 50k steps with top-1 error rate of 35.2%. The training loss curve can be seen in

Fig 4.5.

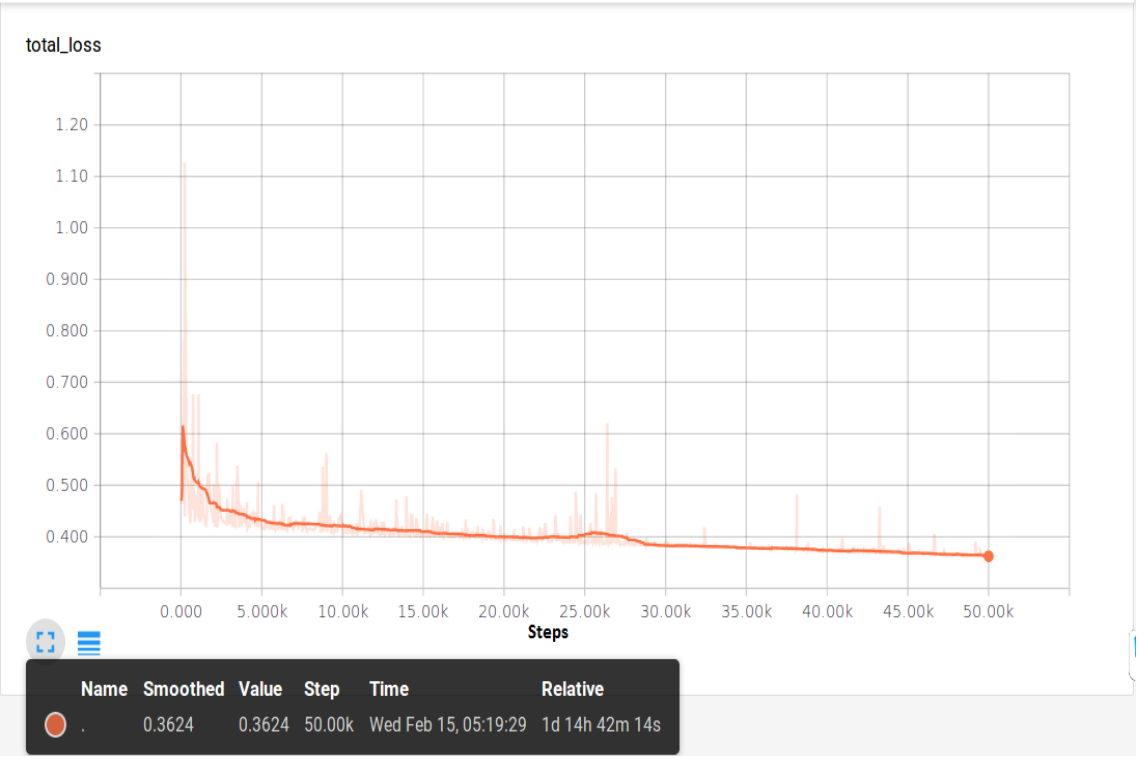

Figure 4.5: Training loss curve of CNN-IR

Table 4.7: CNN-IR Training : Training result and changing of Learning rate

Learning rate	Step	Test Accuracy (%)	Recall @5 (%)
0.04	1k	14.28	29.52
	27,272	81.90	99.04
	30,199	88.57	99.04
	30,615	90.95	99.52
0.02	32,126	92.38	99.04
	33,388	89.04	100
0.035	50k	92.38	100

4.2.4 Proposed CNN-F: Training and Results

The CNN-F network (architecture explained in section 3.4.6) is trained on the augmented data on two Titan X GPUs using Tensorflow, utilizing Tensorflow-slim library. After empirical study, the initial learning rate of 0.003 is used with stochastic gradient descent and softmax cross entropy loss function. The learning rate is reduced by 0.001 whenever a saturated loss

curve is observed. A dropout of 0.8 is used. For comparison, this network is also trained for 50k steps with a batch size of 64, achieving 48% top-1 error rate and 92.61% test accuracy. The training loss curve is shown in Fig 4.6 and the training evaluation is shown in Table 4.8.

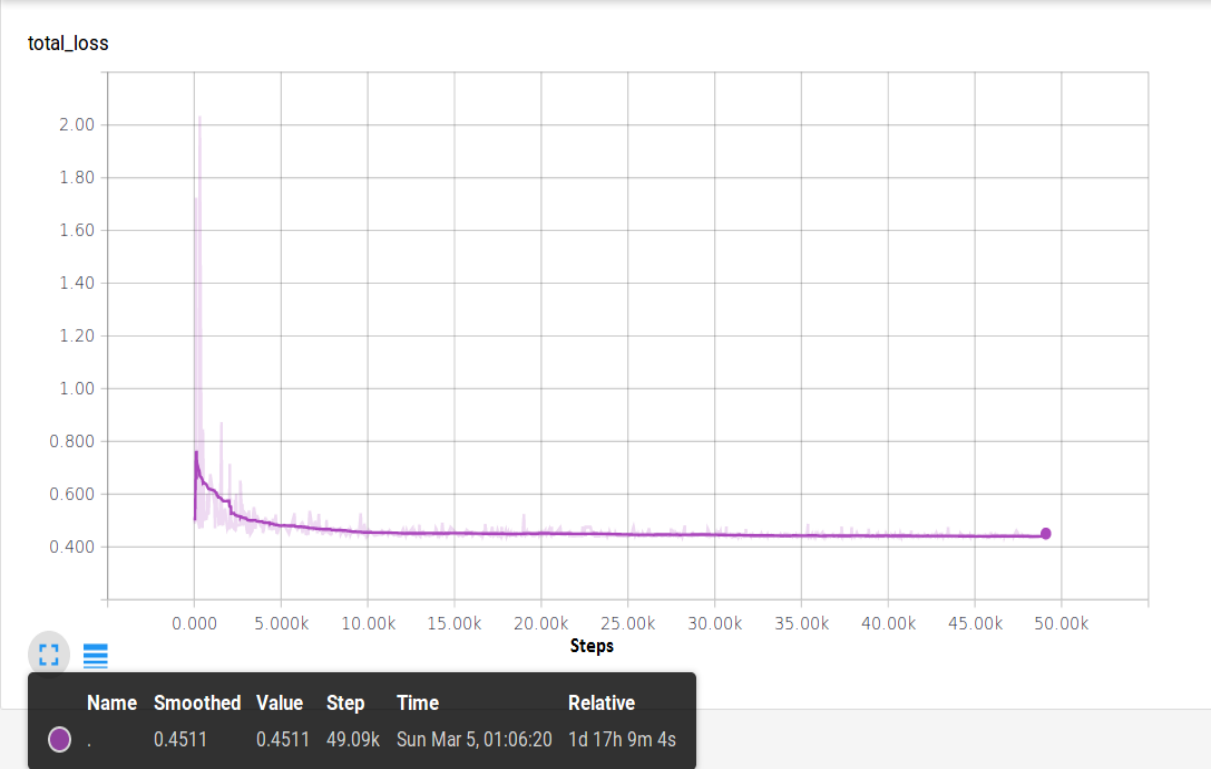

Figure 4.6: Training loss curve of CNN-F

Table 4.8: CNN-F Training : Training result and changing of Learning rate

Learning rate	Step	Test Accuracy (%)	Recall @5 (%)
0.003	11k	63.80	90
0.0035	16k	73.33	97.61
0.0025	17,614	79.52	98.88
	18k	78.51	98.09
0.002	30k	88.57	98.09
	35k	87.38	98.09
0.0038	36,483	90	99.04
0.004	38k	89.28	98.09
0.0048	38,500	85.71	98.88
0.0020	40k	91.42	98.09
	50k	92.61	98.09

4.2.4.1 K-Fold Cross-validation

Cross-validation or rotation estimation, is a validation technique used to do a statistical analysis of the data and to avoid overfitting the model on train data [54]. K-Fold cross validation is a non-exhaustive cross-validation technique, where the original data are divided into K equal sized subsamples. In each rotation, each subsample is retained as a validation set and the remaining $K-1$ subsamples are used for training the K models.

A common choice for K-Fold cross validation is K=10. To make a tradeoff between the time and computational cost, K value = 5 is chosen. The augmented data is randomly split into 5 sets and the training follow the training parameters of the CNN - F and are trained for 50k steps. The following table 4.9 shows the results for each set as a validation set. Figure 4.7 shows the boxplot for the cross validation results.

Table 4.9: Results of 5-Fold Cross-Validation

	Val Accuracy (%)	Val Recall (%)	Test Accuracy (%)	Test Recall (%)
Set1	97.70	100	97.14	98.33
Set2	91.99	100	91.20	99.28
Set3	97.75	99.98	97.38	98.81
Set4	92.72	100	93.33	99.52
Set5	94.42	99.96	93.81	98.57

4.2.5 Score Level Average fusion

In this experiment, the output scores of each classifier of CNN-IR, CNN-F and Inception-v3-R+ are normalized between [0,1] and then the average fusion is applied. The average fusion method is explained in section 3.4.7. The table 4.10 shows the results of average fusion of the three models on each class of image dataset.

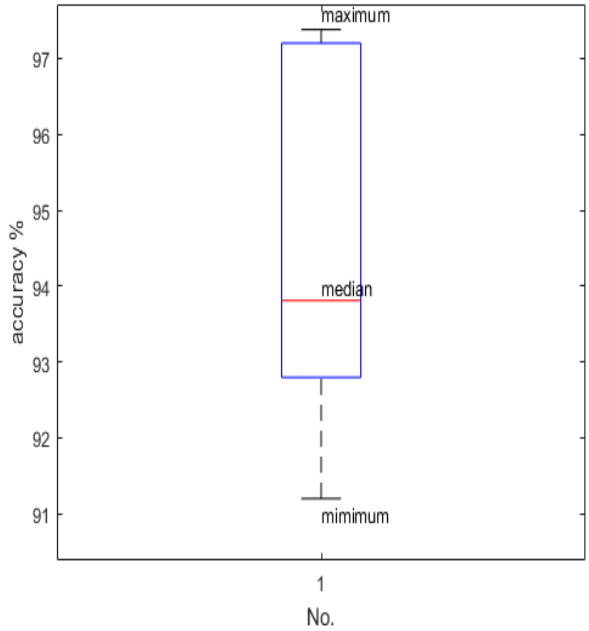

Figure 4.7: Boxplot for 5-Fold Cross validation

Table 4.10: Results of Score level average fusion for Image dataset. Each test set has 30 images

Class	Correctly Recognized	Accuracy (%)
BYA	30	100
BYC	30	100
IOC	28	93.33
KJC	30	100
KWC	30	100
OCB	30	100
OCC	25	83.33
OSC	30	100
RRC	30	100
TFA	30	100
TFC	28	93.33
TLC	30	100
WVA	30	100
WVC	26	86.67
Total	**407 / 420**	**96.90**

4.2.6 Summary of results on Image Dataset

In this section, a comparison table to list the results of all methods evaluated on the image dataset is provided. The results are illustrated in the table 4.11. From the table, it can be seen that the train-time fusion network (CNN-F) with data augmentation performed better than other networks. The average fusion of the three networks has given the highest accuracy of 96.90% on the test.

Table 4.11: Summarizing results of the methods evaluated on the Jersey XIV image dataset

Method	**Input data**	**Average Test Accuracy (%)**	**Best Test Accuracy (%)**
HOG Features+ SVM	original, grayscale	49.05	49.05
Bag of Words	original, grayscale	56	59
CNN-s Softmax	orginal, grayscale	80.79	84.28
CNN-s + SVM (raw)	original, grayscale	84.10	85.00
CNN-s + SVM (hard normalized)	original, grayscale	83.69	83.81
CNN-s + SVM (soft normalized)	original, grayscale	83.88	85.24
CNN-s + KNN (raw)	original, grayscale	82.32	83.33
CNN-s + KNN (hard normalized)	original, grayscale	81.43	82.38
CNN-s + KNN (soft normalized)	original, grayscale	81.96	82.38
CNN-s-r	augmented, RGB	86.67	86.67
Inceptio-v3-Ro	original, RGB	88.27	88.27
Inception-v3-R+	augmented, RGB	92.38	92.38
CNN-IR	augmented, RGB	91.42	91.42
CNN-F	augmented, RGB	**92.61**	**92.61**
Average Fusion (CNN-IR + CNN-F + Inception-v3-R+)	augmented, RGB	**96.90**	**96.90**

4.3 Part-3: Application in videos

4.3.1 Video dataset Test Results

The 14 videos are tested for 4 models : CNN-IR, CNN-F, Inception-v3-R+ and the average fusion of the three models. The video dataset consists of 14 videos with 3,584 frames, of which 2,188 frames containing the bounding boxes. The results are shown in the Table 5.2. The table

consists of a number of frames and bounding boxes (bboxes) for each class, and respective results when each network is applied. From the results, the average fusion methods performed the best with 81.31% accuracy.

Table 4.12: Results of tests conducted on video dataset using all the trained models. One video for each class.

Class	No. of Frames	No. of bboxes	Inception-v3-R+	CNN-IR	CNN-F	Average Fusion Model
IOC	323	181	147	169	172	176
BYC	291	248	164	59	243	178
KJC	242	151	106	140	150	141
KWC	315	214	156	213	49	194
OCB	243	122	83	25	54	54
OCC	268	146	89	144	122	143
OSC	89	89	67	89	46	88
TFA	237	224	166	121	152	209
TLC	256	209	163	96	16	99
WVC	283	109	102	86	63	94
WVA	271	225	212	225	224	225
BYA	181	60	60	45	60	60
RRC	282	66	66	46	66	66
TFC	303	144	117	11	0	52
Total	3584	2188	1698	1469	1417	1779
Accuracy (%)			77.60	67.14	64.76	81.31

4.3.2 No-reference Subjective Image Quality Assessment

Classification constraints for blur, occlusion, good, side pose and all three categories are applied as discussed in the section 3.5.1.1, with the criteria shown in Fig. 3.18. Table 4.13 shows the number of recognized/classified video dataset bounded images fall into the different categories, while Table 4.14 shows categorization of the unrecognized/misclassified images by one human subject.

Table 4.13: Subjective IQA: categorization of recognized/classified images based on considered criteria by human subject

Class	No. of recognized	Blur	Good	Side pose	Occlusion	All three
IOC	176	2	128	9	11	26
BYC	178	15	118	19	22	4
KJC	141	17	98	0	26	0
KWC	194	0	71	123	0	0
OCB	54	1	35	18	0	0
OCC	143	14	62	65	2	0
OSC	88	0	0	88	0	0
TCU	209	53	45	92	0	19
TLC	99	20	46	15	19	0
WVU	94	11	40	34	0	5
WVC	225	6	141	78	0	0
BYA	60	5	0	55	0	0
RRC	66	0	46	20	0	0
TFC	52	8	22	22	0	0
Total	1779	152	852	638	80	54

Table 4.14: Subjective IQA: categorization of unrecognized/misclassified images based on considered criteria by human subject

Class	No. of unrecognized	Blur	Good	Side pose	Occlusion	All three
IOC	5	3	0	0	2	0
BYC	70	51	7	5	0	7
KJC	10	7	3	0	0	0
KWC	20	2	0	17	0	1
OCB	68	21	8	32	0	5
OCC	3	0	0	3	0	0
OSC	1	0	0	1	0	0
TCU	15	5	3	4	3	0
TLC	110	24	41	15	8	22
WVU	15	7	2	5	0	1
WVC	0	0	0	0	0	0
BYA	0	0	0	0	0	0
RRC	0	0	0	0	0	0
TFC	92	15	17	60	0	0
Total	409	135	81	142	13	36

The Graph (see Fig. 4.8) shows the percentage of the categorized: blur, good, side pose, occlusion and all three data present in the overall images, as well as in recognized and unrecognized classes. The results show that the majority of data consists of good quality and side pose images. 52.96% of the blur/low quality of the data are classified correctly and 91.32% of the good quality data are classified correctly. If human observation is taken as a threshold in classifying the data to discard the low quality occluded data, before passing the image to the CNN, the Average-Fusion model accuracy has improved to 86.90% from 81.31%, but at a loss of 16.08% of classified data and 21.48% of the whole data. It can be seen from the bar graph that even though there are high amounts of bad quality images being misclassified, there are few images classified correctly. Example of these recognized and unrecognized images with their categorized labels by a human are shown in Fig. 4.9.

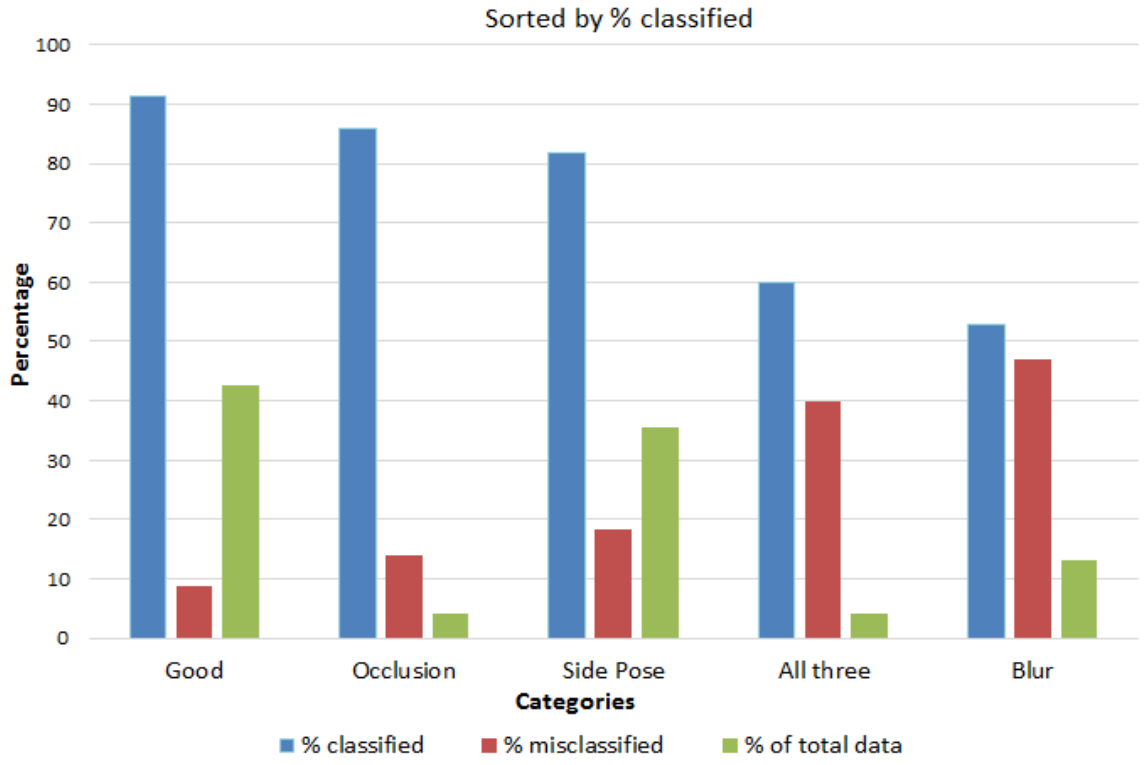

Figure 4.8: Graph showing the percentage of good, blur, side pose, occluded and all three categories in the data, their classified and misclassified percentages. Sorted in descending order of % recognized

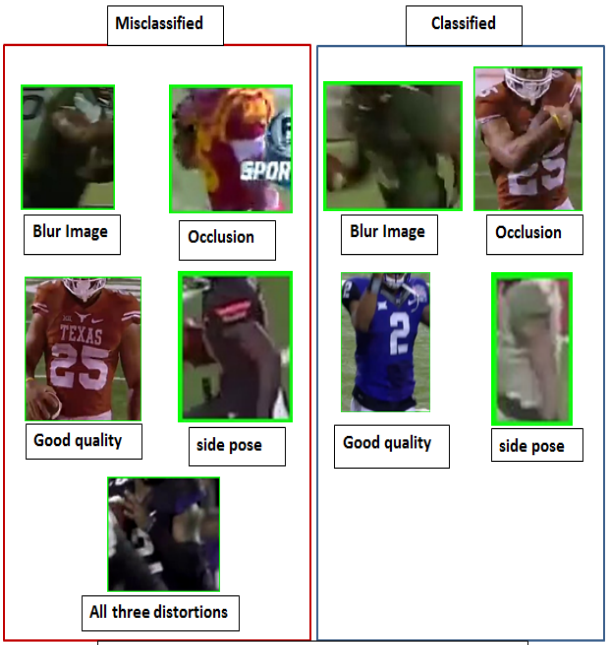

Figure 4.9: Example classified and misclassified images with categorized labels by a human for subjective IQA

4.3.3 Objective IQA: Spatial-Spectral Entropy-based Quality (SSEQ) Index

Publicly available SSEQ software [11] [55] is used to calculate the distortion quality scores of the 2,188 images from the video dataset. All the scores are normalized between [0, 1] for comparison purpose. The higher the distortion quality score, the higher is the distortion. The graph (see Fig. 4.10) shows the distribution of the input data (bounded image per frame) of the video dataset, according to their SSEQ scores, in the range [0,1], with interval size of 0.05.

From this graph (see Fig. 4.10), it is clear that most of the input image data (bounded image per frame) is not of high quality and most recognized data fall under the 0.5 threshold. Most of the unrecognized images are above this 0.5 threshold. To choose a threshold which can increase the performance, a trade off should be made to remove most of the distorted unrecognized images while preserving the recognized images. The graph (see Fig. 4.11) shows the percentage of data under a threshold value and the accuracy obtained at that point. From the graph, a threshold of 0.60 would work as a trade off to improve the performance from 81.31% to 85.23% with a penalty of 27.94% of classified data and 31.26% of the whole data. Few example images with their NIQE scores are presented in Fig 4.12.

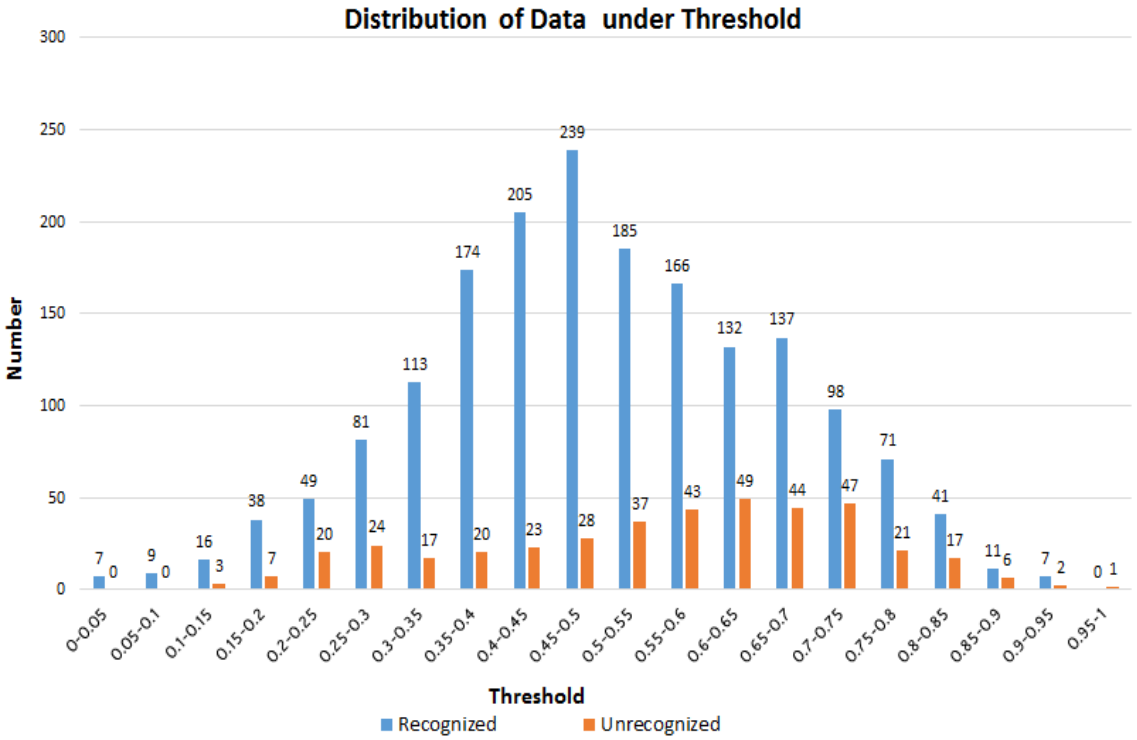

Figure 4.10: SSEQ: Graph showing the distribution of video dataset input images which are recognized and unrecognized by the average-fusion model. The distribution of this data under the SSEQ threshold range of [0,1], with interval size of 0.05 is shown as a bar graph.

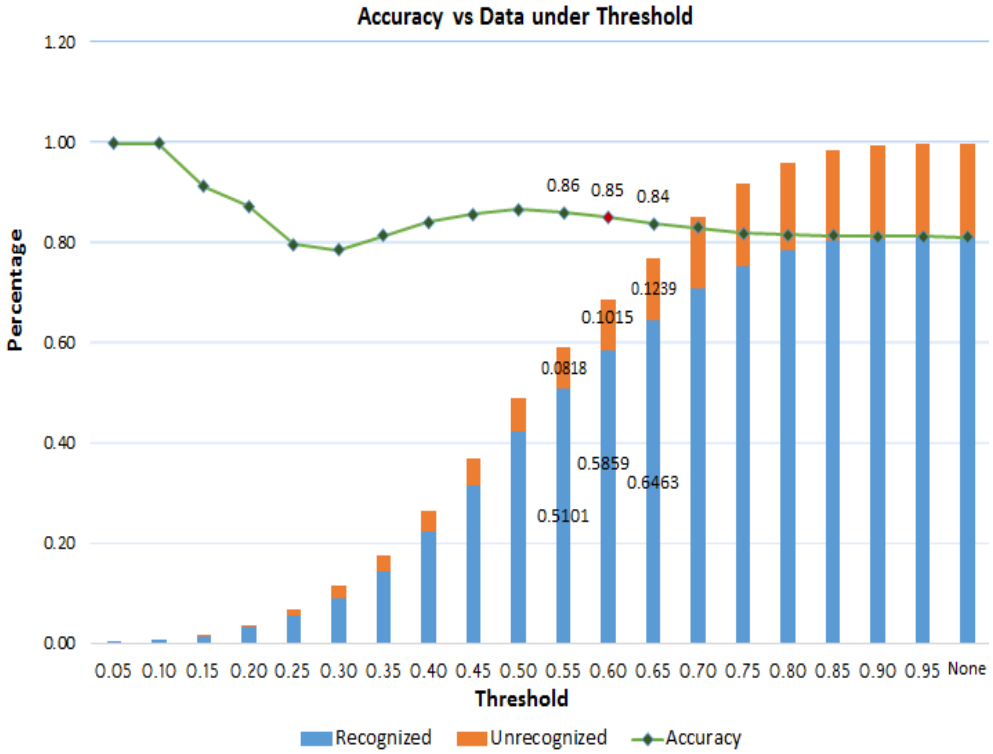

Figure 4.11: SSEQ: Graph showing the % of input video dataset images (recognized and unrecognized by the average-fusion model) that fall under SSEQ threshold value with an interval of 0.05. The line graph shows the % of accuracy at that threshold.

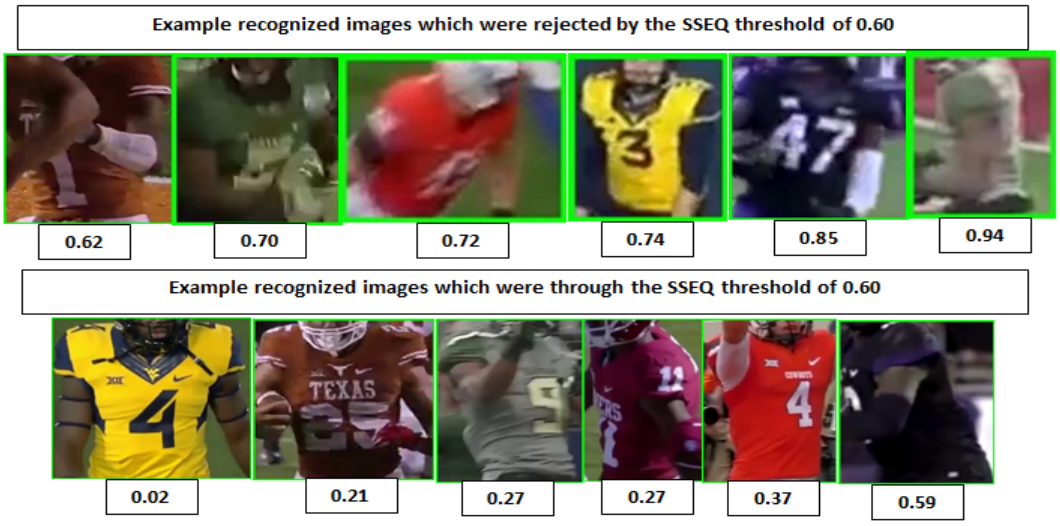

Figure 4.12: Example images of Video Dataset input images and their SSEQ scores.

4.3.4 Objective IQA: Natural Image Quality Evaluator (NIQE)

The NIQE index for the 2,188 images of the video dataset are evaluated using the publicly available NIQE software [56] [12]. All the scores are scaled to [0,1] and as in SSEQ, the higher the score, the greater is the distortion.

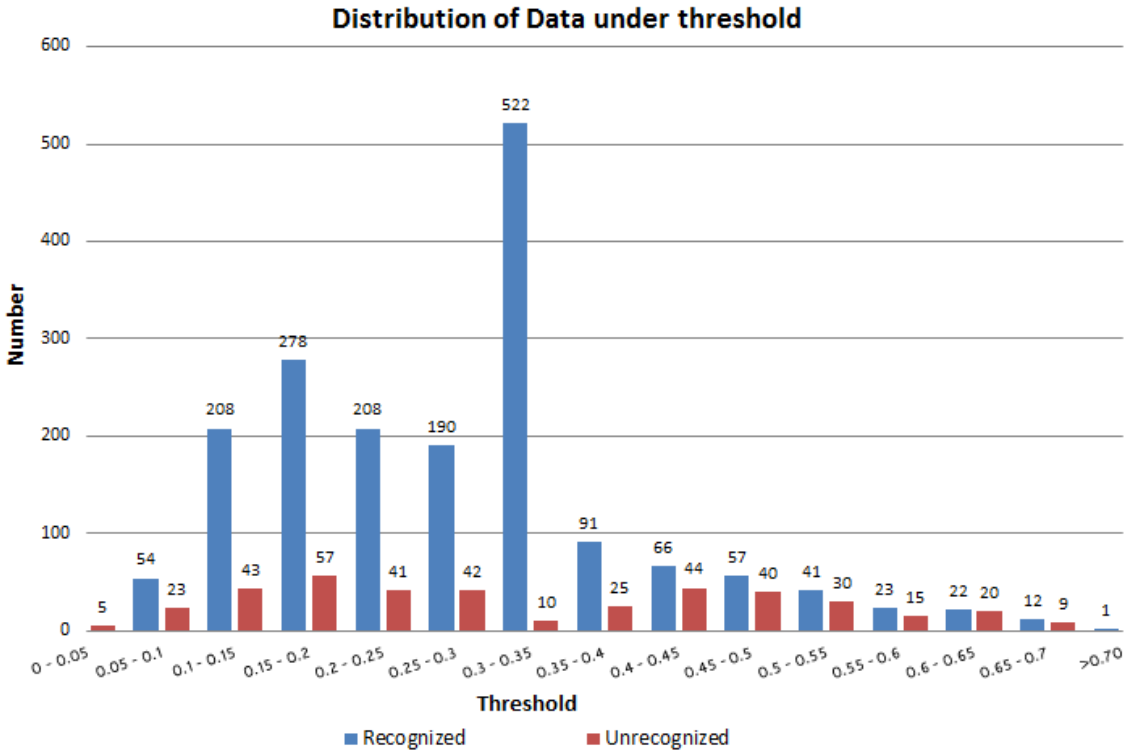

Figure 4.13: NIQE: Graph showing the distribution of video dataset input images which are recognized and unrecognized by the average-fusion model. The distribution of this data under the NIQE threshold range of [0,1], with interval size of 0.05 is shown as a bar graph.

The graph (see Fig. 4.13) shows the input image data (bounded image per frame) of the videodataset distribution, according to their NIQE quality scores, in a range of [0,1] with 0.05 intervals and the graph (see Fig. 4.14) shows the percentage of data under a threshold value and the line graph shows the accuracy obtained at each threshold point.

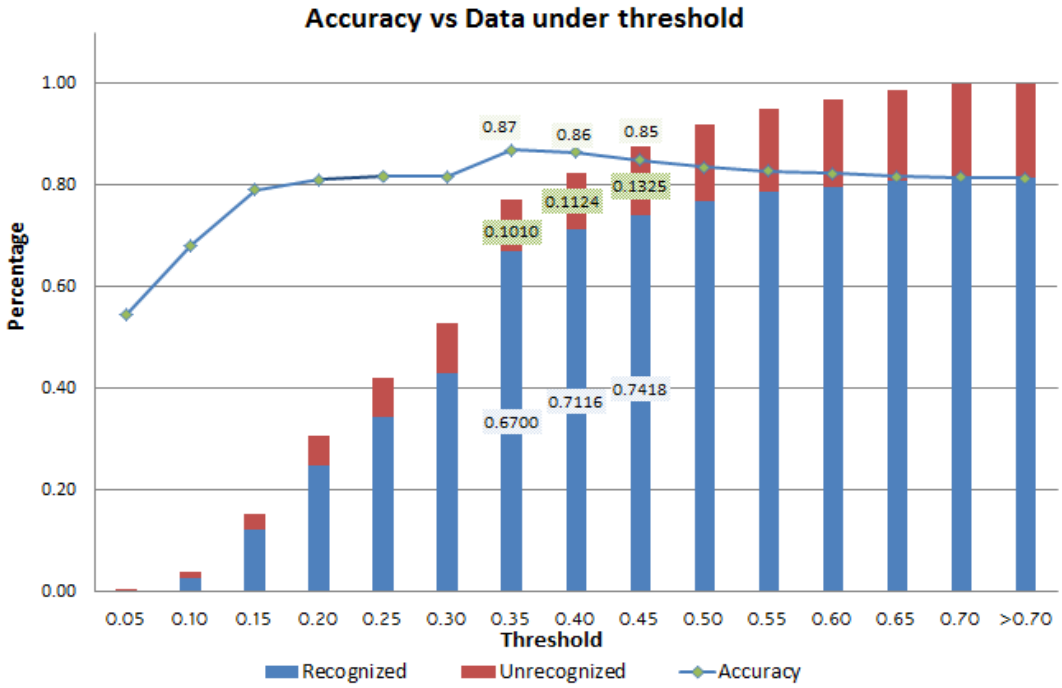

Figure 4.14: NIQE: Graph showing the % of input video dataset images (recognized and unrecognized by the average-fusion model) that fall under NIQE threshold value with an interval of 0.05. The line graph shows the % of accuracy at that threshold.

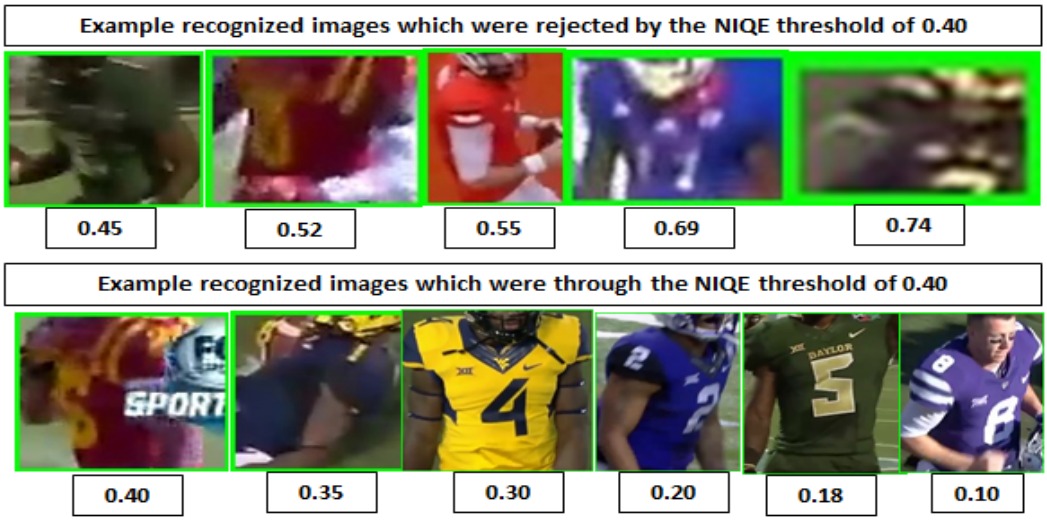

Figure 4.15: Example images of Video Dataset and their NIQE scores.

A trade off can be made at 0.40 threshold value which increases the original 81.31% ac-

curacy to 86.36% with a penalty of 12.48% misclassified data and 17.60% of the overall data. Few example images with their NIQE scores are presented in Fig 4.15.

4.3.5 Summary of IQA Results

In this thesis, three types of Image Quality Assessment are evaluated on the 2,188 images of the video dataset with 1,799 classified/recognized and 409 misclassified/unrecognized images. The goal of the IQA is to filter out any distorted input image due to noise, blur, fast fading or compression. In the Table 4.15 the summary of results for the three methods is presented. The NIQE method performed better with less amount of data loss and improved accuracy of 86.36%, while SSEQ with a threshold of 0.60 penalizes higher amount of data. Even though the Subjective IQA has higher accuracy, it is not practical or favorable for a subjective analysis in real time.

Table 4.15: Summary of IQA methods

Method	Threshold	Accuracy (%)	Penalty of Classified data (%)	Penalty of Overall data (%)
Subjective IQA	None	86.90	16.08	21.48
Objective SSEQ	0.60	85.23	27.94	31.26
Objective NIQE	0.40	86.36	12.48	17.60

Since choosing NIQE threshold at 0.40 increased the accuracy performance of average-fusion model on video dataset to 86.36% with less amount of input data loss, confusion matrices for results of average-fusion model on 14 classes of video dataset before applying the NIQE threshold and after applying the NIQE threshold are provided. Confusion matrix for results by average-fusion before applying NIQE threshold at 0.4 is shown in Fig. 4.16 and after applying threshold is shown in Fig. 4.17. The diagonal of the table shows the number of correctly predicted frames for each of the 14 classes. The test accuracy for each of the 14 classes is also provided in the confusion matrix.

Predict / input	IOC	BYC	KJC	KWC	OCB	OCC	OSC	TFA	TLC	WVC	WVA	BYA	RRC	TFC	Accuracy
IOC	176	0	0	0	0	0	2	0	2	0	0	0	1	0	97.23
BYC	0	178	0	0	0	0	0	50	0	0	0	20	0	0	71.77
KJC	0	0	141	10	0	0	0	0	0	0	0	0	0	0	93.37
KWC	0	1	1	194	0	0	0	0	0	0	0	0	0	18	90.65
OCB	0	5	0	4	54	29	3	7	2	5	0	5	8	0	44.26
OCC	0	0	0	0	0	143	3	0	0	0	0	0	0	0	97.94
OSC	0	0	0	0	0	0	88	0	0	0	0	0	1	0	98.87
TFA	0	0	3	12	0	0	0	209	0	0	0	0	0	0	93.30
TLC	0	0	0	0	0	0	35		99	0	0	36	39	0	47.36
WVC	0	0	0	1	4	0	0	4	2	94	0	4	0	0	86.23
WVA	0	0	0	0	0	0	0	0	0	0	225	0	0	0	100
BYA	0	0	0	0	0	0	0	0	0	0	0	60	0	0	100
RRC	0	0	0	0	0	0	0	0	0	0	0	0	66	0	100
TFC	0	0	30	62	0	0	0	0	0	0	0	0	0	52	36.11

Figure 4.16: Confusion matrix result of average-fusion model on 14 classes of video dataset before applying NIQE threshold of 0.40.

Predict / Input	IOC	BYC	KJC	KWC	OCB	OCC	OSC	TFA	TLC	WVC	WVA	BYA	RRC	TFC	Accuracy
IOC	167	0	0	0	0	0	0	0	2	0	0	0	0	0	98.81
BYC	0	166	0	0	0	0	0	17		0	0	16	0	0	83.41
KJC	0	0	64	0	0	0	0	0	0	0	0	0	0	0	100
KWC	0	0	1	194	0	0	0	0	0	0	0	0	0	18	91.08
OCB	0	1	0	0	53	28	0	1	0	0	0	3	0	0	61.62
OCC	0	0	0	0	0	102	0	0	0	0	0	0	0	0	100
OSC	0	0	0	0	0	0	88	0	0	0	0	0	0	1	98.87
TFA	0	0	3	9	0	0	0	170	0	0	0	0	0	0	93.40
TLC	0	0	0	0	0	0	10	0	92	0	0	2	39	0	64.33
WVC	0	0	0	0	1	0	0	0	0	58	0	2	0	0	95.08
WVA	0	0	0	0	0	0	0	0	0	0	225	0	0	0	100
BYA	0	0	0	0	0	0	0	0	0	0	0	60	0	0	100
RRC	0	0	0	0	0	0	0	0	0	0	0	0	66	0	100
TFC	0	0	30	62	0	0	0	0	0	0	0	0	0	52	36.11

Figure 4.17: Confusion matrix result of average-fusion model on 14 classes of video dataset after applying NIQE threshold of 0.40.

Chapter 5

Conclusion and future work

5.1 Conclusion

In this work, supervised deep learning approaches for the problem of classifying football jersey images were developed and investigated. An image dataset with 14 classes has been created by collecting 7,840 jersey images belonging to 10 teams of Big 12 2015 conference football season. The images were collected from online resources and cropped to center the object of interest using MATLAB GUI. The images vary in terms of pose, standoff distance, occlusion and illumination. The data is augmented 5 times with various levels of Gaussian noise and random shifting and rotating.

Before investigating deep learning approaches, conventional methods such as BOW (Bag of Words) and HOG (Histogram of Oriented Gradient) features were tested, but performed poorly with average test accuracies of 56% and 49.05% respectively. Three CNN models were trained and tested on the image dataset - CNN-s, CNN-IR and CNN-F. Another model is trained using transfer learning of inception-v3 network - Inception-v3-R+. Image dataset is used to train these models. The CNN-s was trained on Matlab using Matconvnet, while other networks are trained using Tensorflow. The CNN-s is trained on the grayscale images achieving 84.28% test accuracy, while when trained on RGB data (CNN-s-r), the same network achieved 86.67%. The CNN-F is a fusion of two networks which are trained simultaneously on the images and are fused at the feature level. This network has achieved the highest test accuracy of 92.61%, while

CNN-IR achieved 91.42% and Inception-v3-R+ achieved 92.38%. After an empirical study of different score level fusion methods, such as product, minimum, maximum and average, an average fusion of CNN-IR, CNN-F and Inception-v3-R+ has achieved the best accuracy of 96.90%.

In this work, since the models were tested on a small set of 420 images, the trained models are tested on a larger scale video dataset. A video dataset is created consisting of 14 videos, one for each class, with 3,584 total frames of which 2,188 frames contain the object of interest. The video dataset presents different challenges than the image dataset, with respect to higher level of occlusion and possible transmission noises. The jerseys are manually specified with bounded boxes in each frame mimicking an object detector. The average-fusion model has achieved an accuracy of 81.31% accuracy.

The model has shown less performance than it is generally expected of. Image Quality Assessment (IQA) is conducted to evaluate if this performance gap is due to the image quality of the videos, which might be subjected to noise and compression during transmission. Subjective analysis is performed by one subject to analyze the classified and misclassified images of the tested average-fusion model. Subjective analysis showed that the highest percentage of misclassified data are blur/low quality data and with a penalty of 16.08% of classified data, it achieves 86.90% accuracy if lower quality data is removed.

Since there is no reference image available for the input images (bounded image per frame), no-reference IQA is needed to be evaluated. For this, two publicly available no-reference IQA analyzing softwares: Spatial-Spectral Entropy-based Quality (SSEQ) Index [11] and Natural Image Quality Evaluator (NIQE) [12] were utilized. Quality scores are calculated and upon evaluation, for SSEQ, a threshold of 0.60 achieved 85.23% accuracy with a penalty of 27.94% classified data. For NIQE, a threshold of 0.40 achieved 86.36% accuracy with a penalty of 12.38% classified data. Considering the trade off between accuracy and loss of input data, NIQE at 0.40 is considered to be a better choice. Even though subjective analysis has achieved a bit better, it is expensive and time consuming to perform subjective analysis in real time.

Proposed System Design:

From the work discussed above, an overall system for classification of jerseys in real-time

on the JerseyXIV Video dataset can be designed as shown in the Fig 5.1. Design steps are taken as per the results of the above evaluations. They are as follows:

- Jersey XIV Image dataset is created to train a convolutional neural network for classification.
- A train-time fusion (CNN-F) network is designed and trained on Image dataset, with a classification test accuracy of 92.61%.
- A score-level average-fusion of the three networks – CNN-F, CNN-IR and Inception-v3-R+ – which are trained on the Image dataset has achieved an overall test accuracy of 96.90%.
- Jersey XIV Video dataset is created to test the models in real-time.
- The average-fusion of the networks has achieved 81.31% accuracy on the Video dataset.
- Image Quality Assessment is conducted to evaluate the quality of Video dataset images (bounding box), as the videos are prone to distortions of noise, compression and fast fading during capturing, transmission and acquiring. A threshold needs to be set to discard any distorted images above it.
- Natural Image quality Evaluator (NIQE) [12] with a threshold of 0.4 on the input images is found to increase the accuracy of average-fusion on the Video dataset to 86.36%, with less amount of data being discarded.

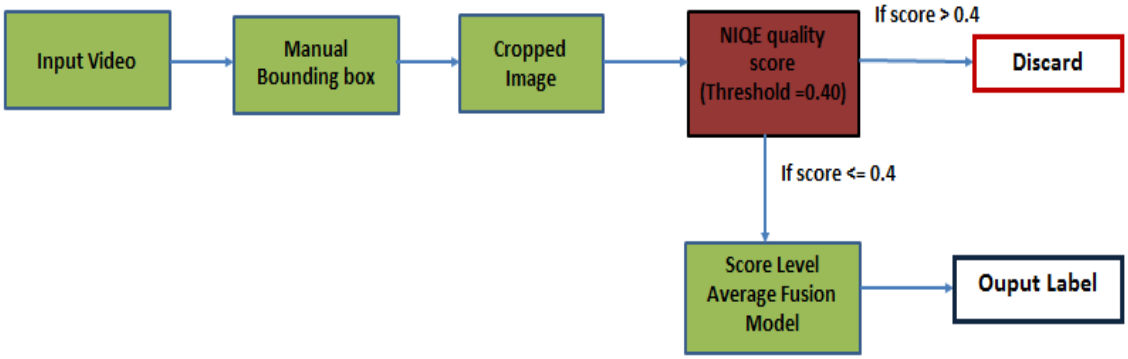

Figure 5.1: Schematic diagram showing the final experimental design process for classification of jersey in a video dataset.

5.2 Limitations and Future Work

The limitations and their corresponding future solutions are as follows:

- More image data can be acquired for training the deep models, since there are proven results that the increased data improve the classification performance and widens the features available for the system to learn. More classes of different upper body clothing can also be added to extend the network's ability in classifying objects in real world images/videos.

- The layers of CNN-F can be increased in depth if more data is available. Very deep neural networks have shown high performance in classification.

- 5-fold cross-validation alone is performed for the CNN-F network due to training time limitation. A more comprehensive study can be made by a 10-fold cross validation.

- Manually bounded boxes per frame are used in the videos to mimic a detector. An automatic detector can be trained to produce an automatic recognition system. This detector can be used to collect more data (and extend the classes) from football games which are held every year. The classification network can be retrained or updated on this data.

- Deep neural networks are considered as a generic algorithm which can be applied to various datasets. The performance of CNN-F can be evaluated on other benchmark datasets (CIFAR-10, ImageNet).

- By collecting or producing synthetic data for blur, noise, fast fading and compression, an automatic classifier for High, Medium and Low classes can be trained on the extraction of the features. This can be used instead of a calculated score for Image quality assessment.

- A football game can be viewed as a binary problem and metadata can be found in the videos or on the field (text, logo). Extraction of these hard features can help in recognition of text or digit to identify the player.

Bibliography

[1] Andrej Karpathy. "neural networks part 1: Setting up the architecture." notes for cs231n convolutional neural networks for visual recognition, 2015.

[2] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[3] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[4] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, and . TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[5] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Lukas Bossard, Matthias Dantone, Christian Leistner, Christian Wengert, Till Quack, and Luc Van Gool. Apparel classification with style. In *Proceedings ACCV 2012*, pages 1–14, 2012.

[7] Rovell Darren. Harris poll: NFL most popular for 30th year in row, 2014.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012.

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] Soren Goyal and Paul Benjamin. Object recognition using deep neural networks: A survey. *CoRR*, abs/1412.3684, 2014.

[11] Lixiong Liu, Bao Liu, Hua Huang, and Alan Conrad Bovik. No-reference image quality assessment based on spatial and spectral entropies. *Signal Processing: Image Communication*, 29(8):856–863, 2014.

[12] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Process. Lett.*, 20(3):209–212, 2013.

[13] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2001.

[14] Farshid Arman and J. K. Aggarwal. Model-based object recognition in dense-range images — a review. *ACM Comput. Surv.*, 25(1):5–43, 1993.

[15] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[16] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[17] David H. Hubel and Torsten N. Wiesel. Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959.

[18] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR*, 2014.

[22] Z. Wang and A. C. Bovik. Modern image quality assessment. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 2(1):1–156, 2006.

[23] Hamid R Sheikh, Alan C Bovik, and Lawrence Cormack. No-reference quality assessment using natural scene statistics: Jpeg2000. *IEEE Transactions on Image Processing*, 14(11):1918–1927, 2005.

[24] Tomás Brandão and Maria Paula Queluz. No-reference image quality assessment based on dct domain statistics. *Signal Processing*, 88(4):822–833, 2008.

[25] Guangtao Zhai, Wenjun Zhang, Xiaokang Yang, Weisi Lin, and Yi Xu. No-reference noticeable blockiness estimation in images. *Signal Processing: Image Communication*, 23(6):417–432, 2008.

[26] Izak van Zyl Marais and Willem Herman Steyn. Robust defocus blur identification in the context of blind image quality assessment. *Signal Processing: Image Communication*, 22(10):833–844, 2007.

[27] Rony Ferzli and Lina J Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). *IEEE transactions on image processing*, 18(4):717–728, 2009.

[28] Anush Krishna Moorthy and Alan Conrad Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513–516, 2010.

[29] Anush Krishna Moorthy and Alan Conrad Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE transactions on Image Processing*, 20(12):3350–3364, 2011.

[30] Michele A. Saad, Alan C. Bovik, and Christophe Charrier. Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE Trans. Image Processing*, 21(8):3339–3352, 2012.

[31] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[32] Daniel L Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517–548, 1994.

[33] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2, 2011.

[34] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[35] Dominik Scherer, Andreas Müller, and Sven Behnke. *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*, pages 92–101. Springer Berlin Heidelberg, 2010.

[36] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress, 2010.

[37] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

[38] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[39] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, and . Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[40] Arnulf B.A. Graf and Silvio Borer. *Normalization in Support Vector Machines*, pages 277–282. Springer Berlin Heidelberg, 2001.

[41] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern Recogn.*, 38(12):2270–2285, 2005.

[42] Hyera Byun and Seong-Whan Lee. *Applications of Support Vector Machines for Pattern Recognition: A Survey*, pages 213–236. Citeseer, 2002.

[43] Weijun li and Zhenyu Liu. A method of svm with normalization in intrusion detection. *Procedia Environmental Sciences*, 11(Part A):256–262, 2011.

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[45] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[46] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, oct 2010.

[47] Christian Szegedy, Wei Liu, Pierre Jia, Yangqingand Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, 2014.

[48] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[49] Dymitr Ruta and Bogdan Gabrys. An overview of classifier fusion methods. *Computing and Information systems*, 7(1):1–10, 2000.

[50] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[51] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multi-scale structural similarity for image quality assessment. In *in Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers, (Asilomar*, pages 1398–1402, 2003.

[52] The Mathworks, Inc., Natick, Massachusetts. *MATLAB version 8.6.0.267246 (R2015b)*, 2015.

[53] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for MATLAB. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 689–692. ACM, 2015.

[54] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143. Morgan Kaufmann Publishers Inc., 1995.

[55] Lixiong Liu, Bao Liu, Hua Huang, and Alan Conrad Bovik. Sseq software release, 2014.

[56] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Niqe software release, 2012.

Appendices

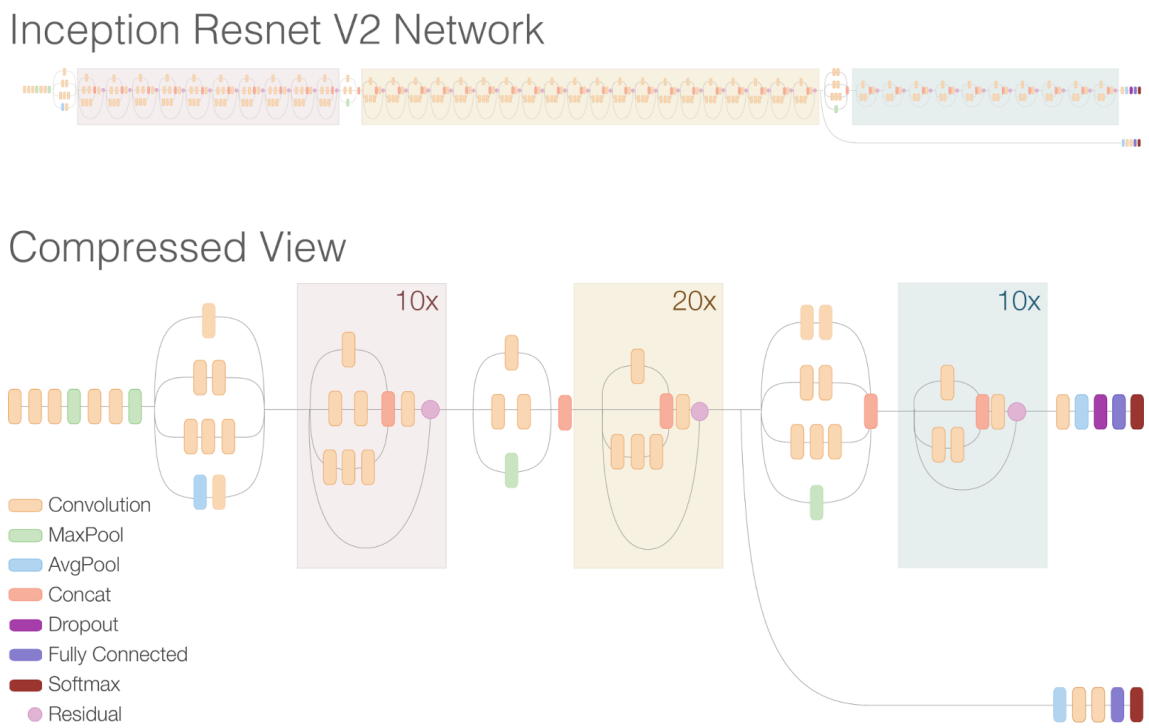

Figure 2: Inception-ResNet-v2 architecture schematic diagram [4]

Table 1: Fusion of CNN-IR, CNN-F and Inception-v3-R+ using the Maximum, Minimum and Product rules. There are 30 test images for each class

Class	Maximum rule	Minimum rule	Product rule
BYA	30	30	30
BYC	29	30	30
IOC	30	28	28
KJC	30	30	30
KWC	20	20	30
OCB	30	30	30
OCC	28	22	18
OSC	29	30	28
RRC	30	30	30
TFA	30	30	30
TFC	15	30	25
TLC	19	30	28
WVA	29	30	30
WVC	6	25	22
Total Accuracy	89	94.05	92.62

Table 2: Average fusion results of CNN-IR, CNN-F set3 model and Inception-v3-R+ on the Video dataset

Class	No. of Frames	No. of bboxes	Average Fusion
IOC	323	181	172
BYC	291	248	170
KJC	242	151	130
KWC	315	214	191
OCB	243	122	64
OCC	268	146	140
OSC	89	89	89
TFA	237	224	171
TLC	256	209	99
WVC	283	109	89
WVA	271	225	220
BYA	181	60	60
RRC	282	66	66
TFC	303	144	50
Total	3584	2188	1711
Accuracy (%)			78.19

Image and Video dataset credits

(i) Image Dataset

1. Getty Images - Ronald Martinez, Ron Jenkins, Rel Del Rio, Alex Menendez
2. `https://www.wacotrib.com/site/terms.html`
3. alomy stock photo
4. nfl jerseys online store – `https://jerseys.manitex25.com/cheap-baylor-bears-25-lache-seastrunk-green-college-football-ncaa-jerseys-for-sale-p-101900.html`
5. Statesman – `https://collegesports.blog.statesman.com/2015/06/23/with-120-uniform-combos-baylor-football-never-will-face-a-what-to-wear-problem/`

6. Uniform Critics – https://uniformcritics.com/football/college/baylor-bears/2013-baylor-white-green-unis/
7. Bleacher report –https://bleacherreport.com/articles/2594696-texas-longhorns-vs-baylor-bears-betting-odds-analysis-college-football-pick
8. SB Nation – https://www.hustlebelt.com/2015/7/22/9008745/belt-loops-uniform-power-rankings
9. Athlon sports & Life – https://athlonsports.com/college-football/ranking-big-12s-2015-football-uniforms
10. Icon sports wire
11. Jerome Miron-USA TODAY Sports
12. KWTX Photo - Austin McAfee
13. Ray Carlin-USA TODAY Sports
14. JP Waldron/Cal Sport Media
15. AP Photo/Tony Gutierrez
16. Tim Heitman-USA TODAY Sports
17. Wikipedia: By Source, Fair use,https://en.wikipedia.org/w/index.php?curid=30628689
18. collegefootball.ap.org
19. 247 sports
20. CBSSPORTS.com
21. expressnews.com
22. www.widerightnattylite.com

23. kusports.com
24. whotv.com
25. newsok.com
26. foxsports.com
27. sbnation.com
28. storminspank – https://s29.photobucket.com/user/storminspank/media/farks/bff.jpg.html
29. Selling sites: Lids — Locker Room by Lids, FansEdge, Fanatics, Eastbay, Kohl's, College football store
30. datemplate.com
31. tajerseys.com
32. stock photos
33. usatoday.com
34. Tim warner/Cal sport media
35. amarillomom.org
36. popscreen.com
37. longhornplanet.com
38. nike.com
39. Wikipedia By Osupdt24 - Own work, CC BY-SA 3.0,https://commons.wikimedia.org/w/index.phpcurid=29778402
40. Bleacherreport.com
41. photos.stwnewspress.com

42. kuathletics.com

43. Iowa State Cyclones vs Baylor Bears 10-24-2015 – https://www.youtube.com/watch?v=J2ee0aEHZHE

44. Baylor Bears vs Kansas Jayhawks 10-10-2015 – https://www.youtube.com/watch?v=kcW-FG_F23Q

45. September 26, 2015 - #24 Oklahoma State vs. Texas – https://www.youtube.com/watch?v=RSf_FWOaV7M

46. TCU at Oklahoma State football 2015 – https://www.youtube.com/watch?v=oSIZmhpbo8Q

47. Oklahoma State Cowboys - Kansas Jayhawks 24.10.15 –https://www.youtube.com/watch?v=91TSXEWNhus

48. TCU Horned Frogs 2015-2016 Pump Up - Summers Productions – https://www.youtube.com/watch?v=YjTfFzZvDqY

49. 2014 Minnesota at TCU – Frogs O' War – https://www.youtube.com/watch?v=2ed_GAEAQlM

50. OU vs Akron 2015 stonecoldsooner –https://www.youtube.com/watch?v=Qk8lvn5vIas

51. Oklahoma Sooners Kansas State Wildcats 17 10 15 – Mutasj – https://www.youtube.com/watch?v=WC8x9OLFs6Y

52. NCAAF 09 26 2015 Maryland at West Virginia 720p – WVURxMan – https://www.youtube.com/watch?v=xzrN08PIFOA

53. West Virginia Mountaineers - Kansas Jayhawks 21.11.15 – Mutasj – https://www.youtube.com/watch?v=DUvkFH5PKb4

(ii) Video Dataset

1. Iowa State Football Highlights vs. Kansas (courtesy FSN) -Cyclones.tv – https://www.youtube.com/watch?v=xLoeumHDlDw

2. Baylor Football: Highlights vs. North Carolina - BaylorAthletics – https://www.youtube.com/watch?v=IKHEjBUGbYQ

3. Ohio at Kansas — 2016 Big 12 Football Highlights -Big 12 Digital Network (Kansas Jayhwaks) – https://www.youtube.com/watch?v=p9tGEK-Do30

4. K-State Senior Day -TheWichitaEagle – https://www.youtube.com/watch?v=kx0N4voALH4

5. TCU at Oklahoma State — 2015 Big 12 Football Highlights-Big 12 Digital Network – https://www.youtube.com/watch?v=oz8S-hFiQwc

6. Kansas at Oklahoma State — 2015 Big 12 Football Highlights – https://www.youtube.com/watch?v=16qpFit7P7g

7. Dede Westbrook For Heisman HD - hambone694 – https://www.youtube.com/watch?v=M0xQn4PbtQY

8. TCU Football 2015 Summer Video - Brandon Cundith –https://www.youtube.com/watch?v=pTKfPL50U1Q

9. Texas Longhorns Football 2015 Season Highlight Tape - Recruit Edits – https://www.youtube.com/watch?v=BF320SJ2XCA

10. Georgia Southern at West Virginia — 2015 Big 12 Football Highlights - Big 12 Digital Network – https://www.youtube.com/watch?v=PVrL9ZNoLh8

11. Oklahoma State at West Virginia — 2015 Big 12 Football Highlights - Big 12 Digital Network –https://www.youtube.com/watch?v=fhynMt8ytQo

12. Texas Tech vs Baylor — 2015 Big 12 Football Highlights - Big 12 Digital Network – https://www.youtube.com/watch?v=OhO9DktVuuo

13. TCU QB Trevone Boykin 2014-2015 Ultimate Highlights - TexasTube – https://www.youtube.com/watch?v=cSau0z9QDqU

14. Texas Tech vs TCU 2015 Football Highlights- Marc Daniel –https://www.youtube.com/watch?v=byjxv7qDM8g