ACTION RECOGNITION IN STILL IMAGES USING VISUAL SCANPATHS

Thesis submitted to
Indian Institute of Technology Kharagpur
for the award of the degree

of

Master of Technology
in
Visual Information and Embedded Systems

by

Dishant Satuley
(20EC65R03)

Under the guidance of

Dr. Debashis Sen

DEPARTMENT OF ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
APRIL 2022

Department of Electronics and Electrical Communication Engineering

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Certificate

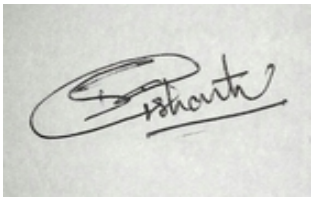
This is to certify that the Progress Report entitled "Action Recognition in Still Images using Visual Scanpaths" submitted by Dishant Satuley (20EC65R03) to Indian Institute of Technology, Kharagpur, India, is a record of bonafide thesis work that has to be done by him under my supervision and guidance in order to get considered for the award of Degree of Master of Technology in the Department of Electronics and Electrical Communication Engineering with Specialization in Visual Information Processing and Embedded Systems.

Dr. Debashis Sen (Project Guide)
Assistant Professor,
Dept. of Electronics & Electrical Comm. Engg.,
IIT Kharagpur

Date: 1st may . 2022

Acknowledgment

I would like to thank my guide Prof. Debashis Sen, my Co-guide Prof. P.K. Biswas and Prof. Saumik Bhattacharya for their valuable time that they gave in providing me a proper feedback on my work. Their suggestion helped me think in a proper direction and go beyond my potential. Be it Prof. Sen's feedback on a good literature survey or Prof. Saumik's suggestion on how to present an idea properly, are invaluable to me and it's because of their suggestions only that I could reach to some positive results in my project. Prof. Biswas corrected my approach many times and helped think out of the box.
I would also like to thank Mr. Ashish Verma (Research Scholar) who constantly supported me during this project and had a brain-storming with me to find some better approach that may help us beat the state-of-the-art accuracy. He gave me the direction to work on the problem.



Dishant Satuley
20EC65R03
M.Tech (VIPES)
Indian Institute of Technology, Kharagpur

Chapter 1

Abstract

Action Recognition in still images has been a challenging task because unlike videos, spatio-temporal features cannot be used here. But Human gaze behaviour can be harnessed to incorporate the temporal information in the task of automated action classification in still images. In this thesis, we have proposed an LSTM based context module that can learn the sequence of object proposals in which they are being observed in a particular scene and based on this learned sequence a still image can be classified into one of the action classes. A sequencer algorithm reorders the object proposals in the sequence provided by human gaze behaviour and before feeding to the LSTM, positional encoding is concatenated with each instance appearance feature. LSTM is capable of learning the sequence in which each instance is being observed and it can also learn the relative position of the current object with respect to the last object in the sequence and thereby remembering the geometrical distribution of the object proposals in the scene. Proposed model is a concatenation of the residual network and and an LSTM based context module and has shown a mean average precision of 92.00% on action classification of still images from PASCAL VOC-2012 action dataset. Residual networks with different depths are experimented and it is found that as the depth of the feature extractor network is increased, the mean average precision is also increased.

Keywords: Action Recognition, Visual Scanpaths, Human Object Relation Network, PASCAL VOC 2012

Contents

List of Figures

Chapter 2

Problem Definition

In case of still images, temporal information is absent and it cannot be used for the classification or detection task. Research in the visual cognition has revealed that a human instead of focusing on entire scene at once, always focuses sequentially on different parts of the scene to extract the relevant information [2]. An LSTM module can be used to learn this sequential information and classify the scene into one of the action categories.

Object Proposals in a scene can be reordered in a sequence provided by the human gaze and this sequence can be learned by the LSTM module. In addition, positional encoding added with each object proposal can help LSTM learn the geometrical position of each object in the scene and hence geometrical distribution of objects in an image can lead to a particular action category.

Chapter 3

Introduction

The task of action recognition has gained quite a momentum in the area of computer vision in the recent years. Videos and images are very strong forms of communication as they can convey a vast amount of information but still understanding the images remains a very complex as well as interesting task that only humans can conduct reliably. Eye movement is linked to how people interpret visuals and has been researched extensively in cognitive science. Several academics are currently incorporating eye movement data into automatic computer vision algorithms to improve picture and video comprehension. Many common computer vision tasks, such as object detection, image segmentation, face recognition, and text detection, employ human gaze. Action detection is one of the new paradigm that is being explored with the gaze data as a tool. We can conduct gaze-enabled action categorization in a variety of applications, including picture retrieval, real-world action detection, and so on[3].

Action recognition in still images is a less investigated area than video-based action recognition. As estimation of motion in still images is not possible, spatio-temporal features cannot be used in this case for characterizing the action. There are several activity types that may be shown in a single frame picture, and these motions can be easily comprehended using human vision. These action categories can be accurately classified by a single image frame. This fact is clear in favour of developing computational techniques for automatic action recognition in still images. [12].

The primary difficulties for action detection in still images include the loss of spatiotemporal signals, backdrop clutter, large intra-class variance and small inter-class variance among specific action classes, change in background illumination, and changes in human posture. The most essential element used to characterise actions in videos is spatial-temporal features. The temporal information is lost in images, making it far more difficult to depict an action [10].

3.1 Different Methods for Action Recognition

In this thesis, we present a deep learning network that learns the sequence in which each item in an image is examined by utilising human gaze behaviour.This method incorporates the temporal information exhibited by the fixation points of human gaze behaviour in the scene action recognition. Other than this method various other methods have been used in the past for action recognition. Let's have a look at them.

Holistic Methods: Holistic techniques capture characteristics from the person in the specified bounding box and blend them with context features of the entire image to recognize human action. To infer human actions, early efforts used a graphical depiction of the human body position. The network presented in [13] that concatenates features derived from a bounding box with features from the entire image for action prediction is one such network that follows this method. Overall, holistic approaches take the simplest approach and do not require many pre-processing stages.

Part Based Methods: Part-based methods look for multiple bounding boxes on different body parts and combine them with global features to anticipate actions. Gkioxari et al. [4] use a sliding window method to train body part detectors on 'pool5' characteristics, then combine these with the ground-truth box to train a CNN for action categorization. To infer human action, Zhao et al. [14] utilize mid-level body part actions (e.g., head: laughing). However, in both the training and testing stages, this system requires an external human posture estimate mechanism to locate body keypoints and clip off portion patches. Furthermore, "hard-coded attention" restricts the areas that can be surrounding a human.

Context Based Methods: Contextual algorithms make use of indications from the surroundings, such as interactive objects. To find appropriate interactive objects, R*CNN uses selective search [4] to create object proposals. These ideas, on the other hand, are required for both the training and testing stages, and sampling over potential proposals could be computationally expensive. Furthermore, R*CNN defines the overlap between the person bounding box and the proposal box using two hyper-parameters.

Weakly-supervised localization: All of the aforementioned techniques need previous knowledge of the ground-truth bounding boxes in both the training and testing stages, making them difficult to scale to real-world applications. A number of recent studies have looked into soft attention or weakly-supervised object localization [15]. Oquab et al. [15] use picture classification to transmit mid-level visual representations to action recognition. Zhang et al. [16] use a five-step iterative optimization method to create a foreground action mask, then extract features from the action mask for recognition.

This approach, however, has a significant optimization complexity. Girdhar and Ramanan [19] suggest a pooling approach that uses a saliency map to scale the score

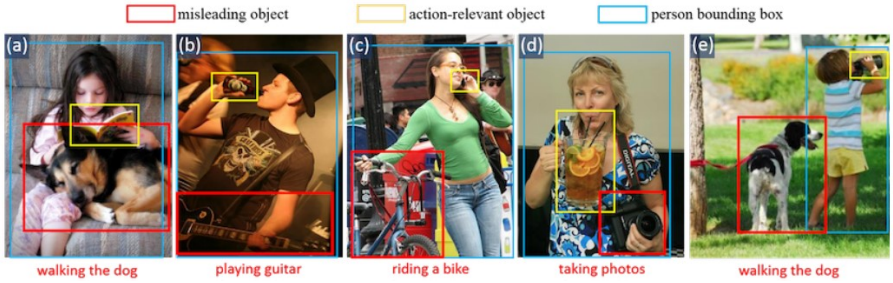


Figure 3.1: Irrelevant objects that cause human action predictions to be incorrect. [10]

map. This strategy may imply that the most relevant cues for recognising actions are the conspicuous items. There may, however, be important but irrelevant objects (see fig. 3.1) that may lead to incorrect predictions.

Multitask Learning: Previous research has demonstrated that learning numerous activities that are related to each other together improves individual performance on all tasks. HyperFace [17], for example, increases individual performance by concurrently learning face detection, landmark localization, posture estimation, and gender recognition tasks. By jointly learning two video datasets, Simonyan and Zisserman [18] used multi-task learning to reduce over-fitting.

Network proposed in this thesis uses the contextual information to decide the action being performed in the image. The object proposals with their respective geometrical positions are learned by an LSTM module, the output of which is used to fuse the score for action classes.

3.2 Applications of Image Action Recognition

Action recognition based on still images has a wide range of applications. It can be used for surveillance, robotics, annotating images with verbs, searching an image database with verbs, searching images online based on action queries, frame tagging, searching in videos, understanding the functionality of an object, and video frame reduction in video based action recognition. Long video sequences can be broken down into lesser frames for action representation, reducing unnecessary data while maintaining accuracy.

Other significant computer vision tasks such as object recognition, video-based action recognition, pose estimation, scene recognition, and image retrieval are all connected to action detection in still images.In numerous applications, such as image retrieval and video based action recognition, still image based action recognition is the first step. The results of still picture based action identification are used as a feature and combined with other extracted features depending on the issue statement. On the other hand, object recognition is used as a precursor to action recognition. The class labels of all the things in the image are typically collected

using off-the-shelf object detectors, and their co-occurrence is modelled and utilised as a feature in still image-based action categorization. Other computer vision challenges, such as posture estimation and scene interpretation, are connected to still image activity detection. The action recognition model is trained with pose and scene information to a given degree of precision. The trained still picture action recognition model is then sent back into the system to help with posture estimation and scene comprehension [12].

3.3 Action Recognition Methods Categorization Based on Features

Due to the lack of temporal information in still picture action detection, traditional spatio-temporal characteristics cannot be applied. In classic video-based action recognition algorithms, low-level information obtained from space-time are important in action recognition. In the realm of still picture action identification, however, low-level characteristics extracted from a single image perform poorly.

Human action detection in still pictures has garnered a lot of interest in recent years because to its complex nature and utility in applications such as image search and retrieval, image annotation, video summarization, and human-computer interaction, to name a few. In a comprehensive assessment, existing action recognition systems were classified based on low-level characteristics and high-level signals used for still image-based action identification. In this part, we'll go over high-level cues and low-level characteristics.

3.3.1 Low-level Features

Dense sampling of scale invariant feature transform (DSIFT), histogram of oriented gradient (HOG), shape context (SC), and GIST are some of the most common low-level features. To extract low-level features for action analysis from a dense sampling of grey scale photographs, the Scale Invariant Feature Transform (SIFT) method is utilised.

3.3.2 High-level Features

The human body, body parts, action-related things, human object interaction, and the entire scene or context are the most prevalent high-level indications for still image-based action detection. On the basis of still pictures, we offer a number of high-level signals for activity recognition.

Human Body

The human body is an important cue for still image-based action recognition. The human body can be manually or automatically recognised in images. For example,

Li et al. [20] manually chose and segmented a minimal bounding box containing sufficient visual information for recognising the human body in a still image for action analysis.

Body Parts

When doing distinct tasks, body parts might be more relevant to action execution than the entire human body. Delaitre et al. [8] used the spatial pyramid bag-of-features to mix the findings of a body part detector with additional data. Poselet is derived from body components and may record the key body positions associated with various tasks.

Object

Some actions may necessitate the availability of several things. Researchers have shown that understanding the linked things can aid in recognising the activities that go with them. Prest et al. [21] employ objectness to determine the probability of a patch being an object. Sener et al. [22] recommended that numerous potential object areas be extracted and used in a Multiple Instance Learning (MIL) framework.

Human-Object Interaction

The interaction between persons and objects, i.e. the relative position and angle between a person and the action-related item, is also significant for action categorization in still pictures, rather than portraying the co-occurrence of individuals and things individually. In still photos, the [23] algorithm learns a mixture model of the relative spatial locations between the bounding boxes of the person and the bounding box of the object.

Context or Scene

While the background of a photograph is frequently used to represent the context or scene of a completed action, the entire image can also be used to represent the context or scene for action analysis, especially when the foreground (e.g., people and objects) occupies only a small portion of the still image. Some activities, such as scuba diving and driving on the road, are only performed in specific environments. As a result, gathering information from the action context or the complete scene may be advantageous for still image-based action identification.

As we have derived our idea from video action recognition, let's have a quick introduction of that part too.

3.4 Action Recognition in Videos

Over the last decade, human action analysis has progressed from previous systems that were often limited to controlled scenarios to today's sophisticated solutions that can learn from millions of records and apply to almost all daily actions. Research improvements in action recognition are being achieved at a quicker rate, culminating in the extinction of what was previously good in a short period of time, given the wide variety of applications, from video surveillance to human–computer interaction [24].

Previous attempts to video action detection have always used similar concepts to those used in image recognition. Human activities, on the other hand, consist of constantly changing motions with varied target objects, and distinct items have different looks in different scenarios.

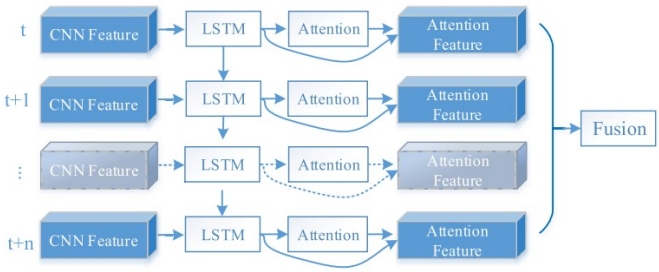


Figure 3.2: Unfolded LSTM with its attention model. [2]

In a variety of sectors, deep neural networks have made substantial progress. These sectors include object detection, recognition, and image classification, thanks to their capacity to learn characteristics from big datasets automatically. Convolution layers of a Convolution Neural Network with orientation-sensitive filters can extract spatial properties of images. CNN may also be used to learn spatio-temporal features for large-scale video categorization by expanding the network's connection in the time dimension. Long Short Term Memory is a recurrent neural network architecture that can extract temporal properties by preserving sequence information over time and capturing long-term dependencies.With the help of the attention model, LSTM has shown promising results in machine translation and picture captioning. So, in our model as well, we used the LSTM model to learn the sequential information.

Chapter 4

Literature Review

Research in computer vision and video understanding has given a rise to the need of large annotated datasets. Annotating such large number of images is a laborious task and practically impossible. This is one of the reasons why automated action recognition in images is becoming increasingly popular. Researchers around the globe have tried and come up with solutions to solve the problem of action recognition in images. In this chapter, we will discuss the different approach to automate the image action recognition problem which only outlines the steps taken by most of the researchers in the past.

Zhang et. al. [**16**] divided the problem of action into two subproblems considering the divide and conquer rule. In the first subproblem, the focus has been on delineating the detailed shape of the human-object interaction region. These regions act as the action mask. In the second subproblem, the focus has been on the proper feature representation for the recognition task.

A combined optimization technique has been presented to obtain a meaningful foreground action mask of the image in the first subproblem. The object proposals are created initially in the input picture using selective search. Following the creation of object suggestions, the creation and depiction of pieces takes place. The foreground action mask is then learned, and the global representation is computed using this foreground mask.

The second subproblem is concerned with the encoding of features for action recognition. Because each action may contain a different number of objects, instead of explicitly stacking component characteristics to generate an action vector, all of the items are fused into an action representation. Only the object proposals $B_{i,j}$ with a suitable overlapping percentage with the action mask α_I in each picture i are considered for encoding the object-based sections. A considerable quantity of extraneous background is filtered out as a result of this.

The set of items in the action mask now represents image activity. Instead of utilising the bag-of-visual-words approach to encode the set of action-related items, an efficient product quantization method was used to retain the same dimension of

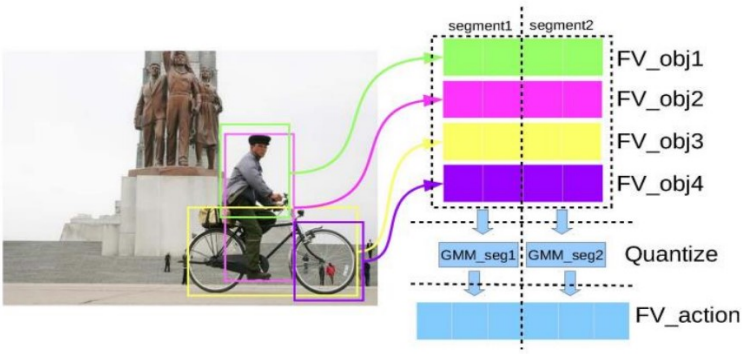


Figure 4.1: Action Representation. *Image reference: Zhang et. al.* [16].

action representation (since various actions may entail varying numbers of objects).

As illustrated in fig.4.1, all the feature vectors from objects of all action classes are first divided into short segments or sub vectors of equal lengths. Then, for each segment, a distinct GMM is learnt. The learnt GMM is then used to compute a fisher vector based on the object characteristics related to the segment of a specific action. The fisher vectors are then concatenated over segments to depict the activity of interest. Finally, a learnt one-vs-all linear SVM classifier is used to predict action class.

Gary Ge et. al. [3] devised a method for action recognition using human gaze behaviour. Use of human eye movement in the action recognition in my opinion has been like hybridization of artificial and human intelligence. Training a classifier with human gaze data to recognize the action being performed in an image is just like helping a classifier with human intelligence.

Gaze data was collected for the PASCAL VOC 2012 action dataset with an SMI iView X HiSpeed 1250 tower-mounted eye tracker [3] for all the 10 classes with the help of 8 observers. These observers were instructed to distinguish activities in a still picture and assign them to one of the PASCAL VOC dataset's ten classes. These participants were given 3 seconds to freely examine a picture, during which time their x- and y-coordinate gaze positions were recorded. The first of these fixations was discarded because subjects began by fixating a cross corresponding to the centre of each image. An example of human eye movement data is shown in Figure 4.2.

For each picture in the dataset, fixation points from all individuals were displayed, with saccades marked by a line and order denoted by the colour temperature. Fixations for all individuals are plotted on the same picture to create aggregated fixation visualisations. More densely packed fixations imply spatial agreement between participants, whereas comparable saccade line patterns indicate temporal agreement. In addition, using the Gaussian Mixture Model GMM, all of the individuals' fixations were grouped. If the fixation clusters are thick, it indicates region of interest with high subject agreement, but if they are sparse, it indicates idiosyncratic variances in viewing behaviour, maybe connected to scene context es-

Figure 4.2: Example of human eye movement data. *Image reference: Gary Ge et. al.* [3].

tablishment. Finally, fixation density maps were created using a two-dimensional gaussian distribution with sigma equal to one degree of viewing angle and weighted by fixation length.

For the quantization of the spatio-temporal data, an image was divided into segments and the features were extracted with help of gaze data and these segments. In an image, different segments may be person, upper body of the person, lower body of the person and the context. Number of transitions between segment pairs, average/max of fixation-density map per segment, dwell time per segment, and measure of when fixations were made on the person vs when they were made on the context were all gaze characteristics extracted from these fixations.

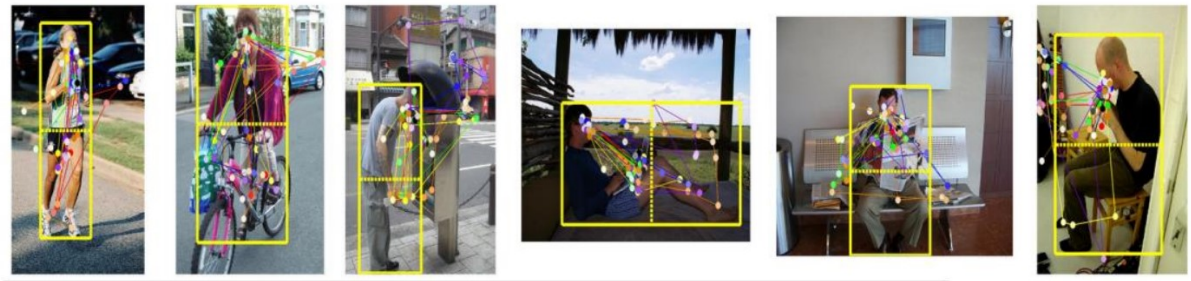


Figure 4.3: Gaze transitions between the upper body and the lower body. *Image reference: Gary Ge et. al.* [3].

The number of fixations and the dwell time over the upper body and lower body [3] was found to be significantly different between different classes as shown in fig.4.3.

Action Classification

For the purpose of action classification, two support vector machine classifiers were trained. Because there was no significant difference between classification with a linear kernel and classification with an RBF kernel for CNN features with fixed subregions, the SVM classifier was trained with a linear kernel as a baseline. For the gaze classifier, a polynomial kernel was used. All features were normalized before training the classifiers. Then a one-vs-all binary classification was conducted

for each action class mean average precision was recorded. To calculate the mean average precision, the average precision for each action was calculated first and then the mean of all the average precisions was taken for all the classes. A confusion matrix was also created for the gaze classifier in order to discover which actions were frequently confused. All of these often misunderstood activities were grouped together, and classifiers were retrained.

Wentao Ma et. al. **[1]** proposed a a Human-Object-Relation-Network that uses the contextual information in the image as a cue to recognize the action. As compared to video action recognition, identifying action in an image is a more challenging task because of the absence of temporal information. As a result, researchers have attempted to combine the various cues with human body characteristics in still photos to more efficiently discern activities, according to [1]. One of the most important signals is surrounding object information, which is employed in a variety of approaches [4].

Figure 4.4: Objects in the surrounding of the human may mislead classifier to identify the wrong action class if relation between human and objects is not computed. Human in question is in blue box and the surrounding objects are in the yellow boxes. *Image reference: Wentao Ma et. al.* [1].

Also, consider that if we don't calculate the relationship between the human and the object, using object characteristics may cause action recognition to be erroneous. We can see in Fig.4.9(a), the bikes around the person may cause him to be misinterpreted as riding the bike but in actual, he is just standing. Similarly, the skateboard and the person standing close in fig4.9(b) may cause the person sitting on the ground to be misidentified as a skate rider.

In this approach, Human-Object relation network is used which is the concatenation of a residual network and human-object relation module [1] which is inspired by the self-attention model. To improve characteristics for action identification, this model utilised the appearance and spatial position of person and object, as well as the pair-wise relation between human and object.

Like other two approaches mentioned above, this model was also trained on the PASCAL VOC 2012 Action dataset. Before looking on the results thrown by this model, let's deeply understand what this model actual does and what is computation procedure for the relation score.

Network Overview
The architecture of the network used in this approach is as shown below.

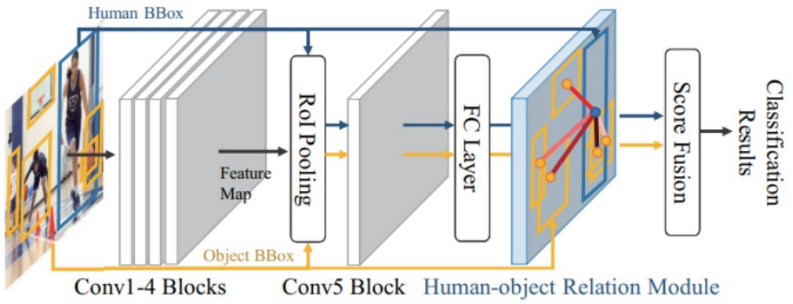


Figure 4.5: Human Object Relation Network Architecture. *Image reference: Wentao Ma et. al.* [1].

The above network is an integration of a feature extractor network, ResNet [26] and human object relation module. As spatial information is also needed, it is required to pass the human and object bounding boxes [1] along with the image in order to compute spatial relation between human and object. Human bounding boxes are available with the PASCAL VOC 2012 action dataset [6] and object bounding boxes are computed through R-CNN [1].

First four convolution blocks of the ResNet compute the image-level feature map. In order to retrieve the instance level features according to human and object bounding boxes, region-of-interest pooling is used on this image-level feature map. The network has now evolved into a two-stream network. The instance-level features are now routed through a fully linked layer and a weight-shared convolution block. The appearance level characteristics for person and object instances are output by this layer. These appearance level features are directly fed to the human-object relation module along with the human and object bounding boxes. Using these parameters as input, relation module computes the relations between human and object instances. These relations are used to enhance the classification scores.

Human-Object relation module In this approach, the aim is to use the relation score to enhance the features that may help recognize the action better. Also, strengthening the object features may help us recognize the actions which seem similar like "riding the bicycle" and "riding the horse". As a result, the human object relation module generates two relation enhanced characteristics: f_{ho} and f_{oh}, which stand for human object relation features and object human relation features, respectively. These relation-enhanced features are further used in our network to fuse with the classification score. As the relation between human and object is naturally mutual and equivalent, a relation weight which is shared between the two and this weight is applied to both human and object features.

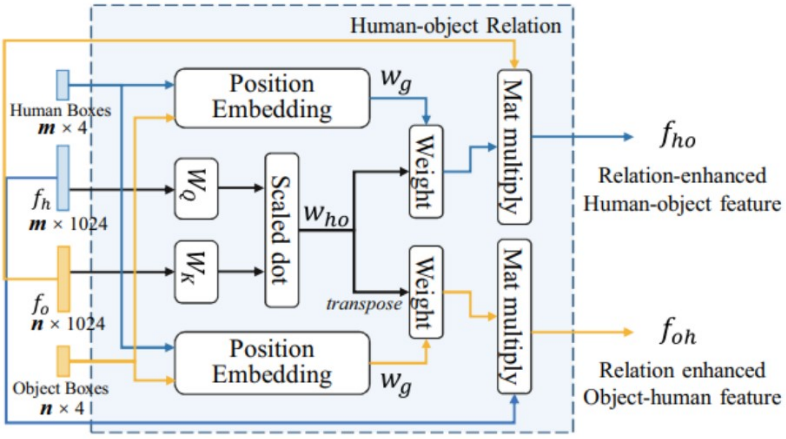


Figure 4.6: Computation process of the human-object relation module [1]. *Image reference: Wentao Ma et. al.* [1].

But we use transpose of this weight shared matrix on one side to meet the dimensionality requirement.

Given the appearance features (f_h and f_o) and human and object bounding boxes as the input, self-attention based scaled dot formula is used to compute the relation weight w_{ho}.

$$w_{ho} = softmax(\frac{w_q f_h . w_k f_o}{\sqrt{d_k}}) \tag{4.1}$$

Where d_k is the number of dimensions of $w_k f_o$ that is utilised as a scaling factor in the training phase to provide a more stable gradient.

However, we may argue intuitively that the human-object relationship is not simply about appearance aspects, but also about geographical positions. As illustrated in fig.4.6, the geometry weight w_g related to position embedding is also determined.

$$w_g = fc(\epsilon_g(b_h, b_o)) \tag{4.2}$$

As now we have the relation weight w_{ho}, the geometry weight w_g and object features f_o, the relation enhanced human-object feature [1] f_{ho} can calculated as following:

$$f_{ho} = fc(\sum_{i=1}^{n}((w_g . w_{ho}) . f_o)) \tag{4.3}$$

But rather than relying on a single human-object relation module, to generate d_k dimensional relation enhanced features, multi-head attention form is used. In the multi-head attention form N_r modules in parallel are used.

Lei Wang et. al. [2] proposed a lightweight action recognition architecture based on deep neural networks that only needs RGB input.

Convolution neural networks (CNNs), long short-term memory (LSTM) units, and a temporal-wise attention model make up the suggested architecture (see fig. 4.7). To begin, the CNN is utilised to extract spatial features that may be used

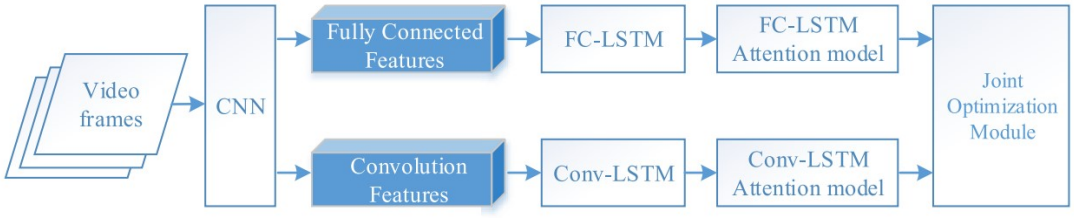


Figure 4.7: Flowgraph of the proposed method. *Image reference: Lei Wang et. al.* [2].

to identify objects from the background using both local and semantic data. The LSTM is then used to create a temporal-wise attention model to learn which parts in which frames are more essential. Finally, a collaborative optimization module is used to investigate the inherent relationships between two types of LSTM features. We have derived the idea of our model from the architecture given in fig. 3.1 and fig. 4.7.

According to **Kaiming He et. al. [26]**, greater the number of layers in a Neural Network can make it highly resilient for image-related tasks. However, it is possible that they will lose accuracy as a result of this. Residual Networks are used in this situation.

Deep learning practitioners have a propensity to add a lot of layers in order to extract key characteristics from complicated pictures. As a result, the initial layers may identify edges, whereas the latter layers may detect recognised objects, such as automobile tyres. However, if the network has more than 30 layers, the performance falls and the accuracy drops. This contradicts the popular belief that adding layers to a neural network will improve its performance. This is not due to overfitting, because in that scenario, dropout and regularisation techniques may be used to eliminate the problem entirely. It's mostly there due to the vanishing gradient issue.

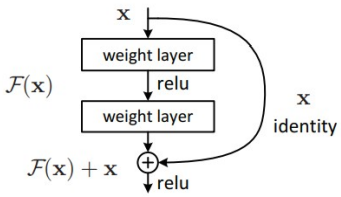


Figure 4.8: Residual learning: a building block. *Image reference: Kaiming He et. al.* [26].

A residual network is made up of identity-connected residual units or blocks (also known as skip connections). In the residual block, the output of the preceding layer is added to the output of the layer following it. The hop or skip might be one, two, or even three times. Due to the convolution process, the dimensions of x may differ from F(x) when adding, resulting in a decrease of its dimensions. To modify

the dimensions of x, we add an extra 1 x 1 convolution layer.

In a residual block, a 3 x 3 convolution layer is followed by a batch normalisation layer and a ReLU activation function. The method is finished with a 3 x 3 convolution layer and a batch normalising layer. The skip connection is added just before the ReLU activation function, bypassing both levels. Repeating such residual blocks creates a residual network.

Chapter 5

Gaze Data

To understand the gaze data better, we first need to understand some key terms like fixations and saccades.

5.1 Fixations

The human eye's spatial and temporal sampling capabilities restricts how we gather visual information from occurrences in the world. We have a repertoire of eye movements that allow us to aim our eyes towards target places of interest since visual acuity diminishes fast as we travel out from the centre of our visual field.

fixation is the period of time where the eye is kept aligned with the target for a certain duration, allowing for the image details to be processed.

Fixations help in understanding the cognitive processes of the human beings. These are the periods when the human eye stops scanning the scene and locks on the centre foveal vision so that the visual system can capture the fine details of what the eye is seeing.

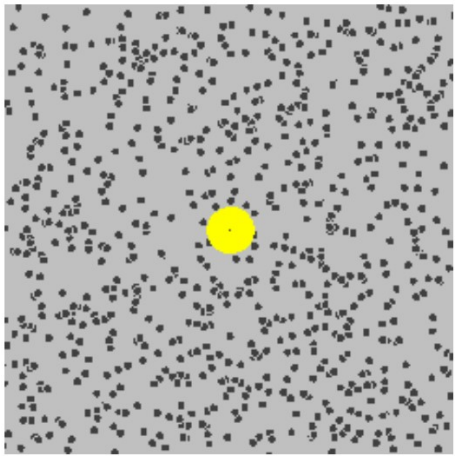


Figure 5.1: The yellow point shows the fixation in the Brownian motion image *Image reference: Wikipedia.*

Gaze points are the instantaneous spatial locations of the visual axis landing on the stimulus. Coordinate (x, y) and the timestamps are used for their measurement.

Suppose that the device is operating at 60 Hz frequency, then the gaze point will be taken at every $1/60 = 16.7$ milliseconds. At the frequency of 300 Hz, gaze points will be separated by 3 milliseconds. So, the gaze points are not the points that can actually be used for the tracking and have no direct relation with the things that the researchers are interested in.

Fixations can be distinguished from the gaze points with its two characteristics.

1. As the fixations are made up of multiple gaze points, they have duration along with the spatial locations.

2. Fixations aren't real in the sense that they can't be measured directly. They're really the result of a mathematical process that turns a succession of raw glance points into a fixation sequence. Fixations are genuine in the sense that they are meaningful appearing events that our visual system generates. The gaze point-to-fixation algorithm, or fixation filter, is meant to simulate the dynamic dynamics of these occurrences.

Fixations have the characteristics that can reveal useful information about attention, visibility, mental processing and understanding. Like if the time taken to make the first fixation is larger, it shows that there is decrease in the salience or visual attractive power of the feature. An increase in the average fixation duration signifies the greater efforts needed to make sense of the scene or else may signify that the scene is more engaging.

Fixations are made up of small, slow movements that help the eye align with the target and prevent perceptual fading. The minimum length necessary to capture information varies depending on the job and stimuli, and can range from $50 - 600 milliseconds$.

5.2 Saccades

Saccades are a type of eye movement that involves quickly moving the fovea from one area of attention to another. Saccades can be freely or involuntarily induced. Saccadic movement occurs when both eyes move in the same direction.

The time it takes to prepare a saccade varies by task and ranges from $100 - 1000 milliseconds$, whereas the typical duration of the saccade is between $20 - 40 milliseconds$. A saccade's length and amplitude are linearly connected, meaning that bigger leaps result in longer durations. When the eye is moving, the end points of a saccade cannot be modified. Saccades do not necessarily follow a straightforward linear path.

5.3 Collection of the Gaze Data

As we have seen in the literature review, fixation data was collected with the help of an SMI iView X HiSpeed 1250 tower mounted eye tracker over the entire PASCAL VOC 2012 Action dataset [3] for all the classes. 8 observers were asked to look at an image for a fixed duration and then label the image as one of the 10 classes as

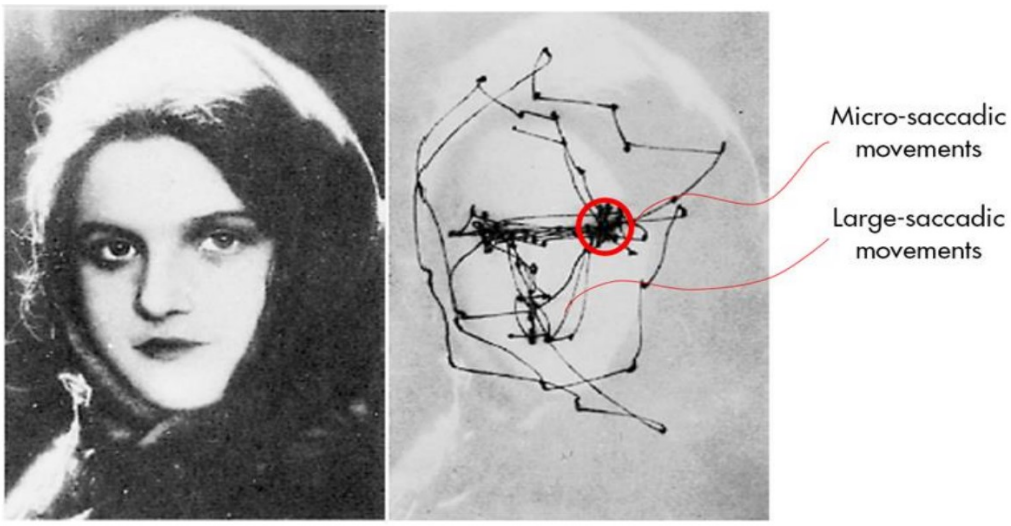


Figure 5.2: Lines on this image show the saccades *Image reference: Wikipedia.*

given in the PASCAL VOC 2012 dataset. During that fixed duration of 3 seconds, x and y co-ordinate of gaze positions are recorded. Because observers began each trial by fixating a cross matching to the centre of each picture, the first fixation point was eliminated.

5.4 Structure of the Gaze Data

Following are the 12 subjects that were involved in the collection of the gaze data. Columns respectively shows their unique ID, their age, their gender, their group and the dominant eye.

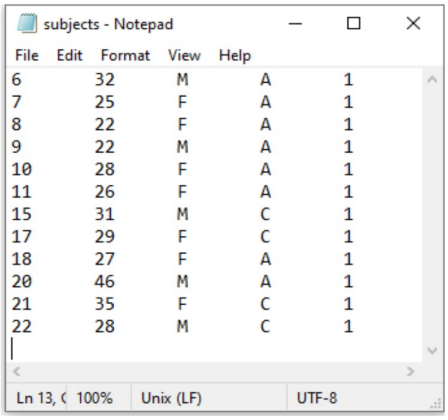


Figure 5.3: Subjects involved in collection of the gaze data.

There are actually two groups of the above subjects. One group was asked to recognize the action in the image and the second group was asked to recognize the context. As we are interested in the action recognition, we have to deal with the group A only.

Following image is the geometry of the setup showing the distance between viewer and screen, screen width, screen height, horizontal resolution and vertical resolution.

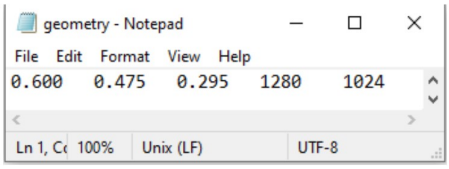


Figure 5.4: Geometry of the setup.

For the image in fig.5.5, the fixation points by different users with user ID 006, 007 and 008 are given in the fig. 5.6.

Figure 5.5: An image from PASCAL VOC-2012 dataset.

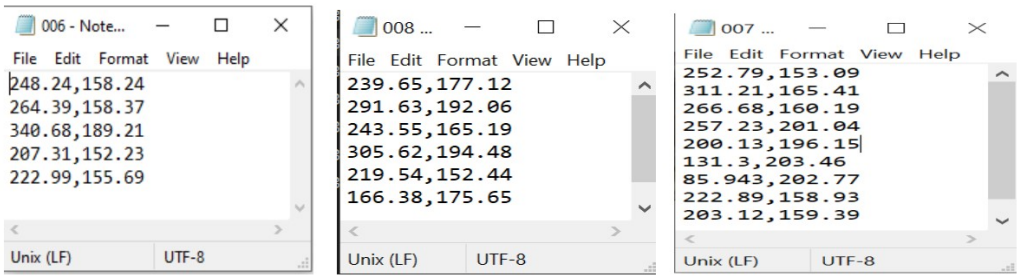


Figure 5.6: Fixation points computed for an image from the PASCAL VOC 2012 action dataset.

Chapter 6

Visual Scanpath-Based Dynamic Context Model for Action Recognition

Human gaze also exhibit temporal information that reveals the pattern in which the different parts of scene are being observed. We can utilize this information for action recognition. Like in videos, next frame can be predicted with the learned sequence of frames; in the same way, we propose an idea of predicting the action label with learned sequence of objects proposals in which they are being observed.

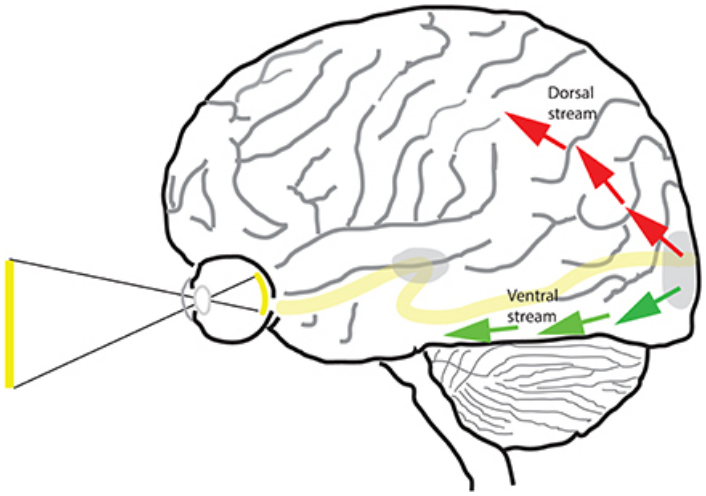


Figure 6.1: Human Visual Cortex System. *Image reference: Wikipedia.*

.

The ventral and dorsal streams of the human visual cortex (see fig. 6.1) are responsible for object and motion recognition, respectively [2]. Humans usually focus progressively on distinct areas of a scene to extract essential information, rather than on the full scene at once, according to visual cognition study. Based on this idea, propose the model as shown in fig. 6.2.

6.1 Model Description

Our proposed model basically consists of three parts, a Convolutional Neural Network (residual Network), A sequencer model and a Long Short-Term Memory based context module. We will look into the working and explanation of each of the part.

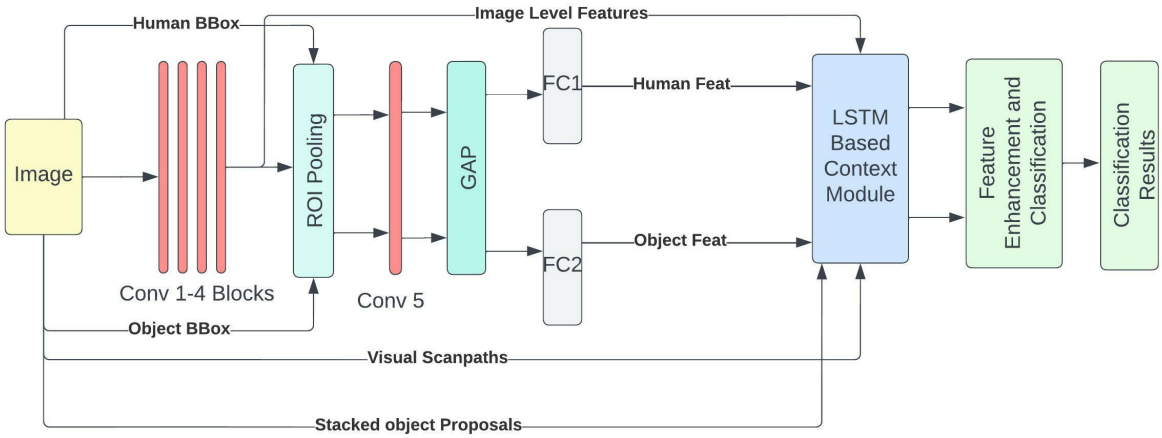


Figure 6.2: Proposed model.

.

Our model is the integration of a Residual Network, a sequencer model and an LSTM based context module. ResNet is basically used for feature extraction and is capable enough to be used in other backbone networks as well [1]. As we want this network to learn the sequential information in which human eyes are tracing the scene, we require human and object bounding boxes as an additional information. Human bounding boxes are provided with the PASCAL-VOC 2012 Action dataset [27] while the object bounding boxes are generated using the Faster-RCNN.

First of all, image level feature map is extracted using the first four convolutional blocks of the ResNet. The number of feature maps depends upon the number of filters in the last layer of the covolutional block. We have used advanced architecture of the Residual Networks [28] in our model. This advanced architecture of the ResNet is available in the Pytorch Image Models Library which is used for image classification, containing a collection of image models, optimizers, schedulers, augmentations and so many other things.

The detailed architecture of the ResNet50 model is given in the following figure 6.3.

Output of the fourth convolutional block is passed to the Region of Interest pooling layer which also takes the human bounding boxes and the object bounding boxes as the input.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 6.3: ResNet50 Architecture.

Region of Interest Pooling

Region of interest pooling is a neural-net layer used for object recognition tasks that speeds up both training and testing significantly. The ROI pooling layer takes two inputs:

1. A fixed-size feature map produced by the last layer of the fourth block of the ResNet and max pooling layers.

2. An N x 5 matrix that represents a list of regions of interest, where N is a number of RoIs.

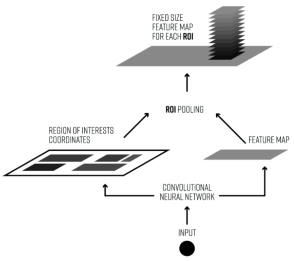


Figure 6.4: Region of Interest Pooling Pipeline.

.

Take a look at a simple example to discover how it works. We'll pool regions of interest on a single 8-by-8 feature map with one region of interest and a 2-by-2 output size. This is how our input feature map looks:

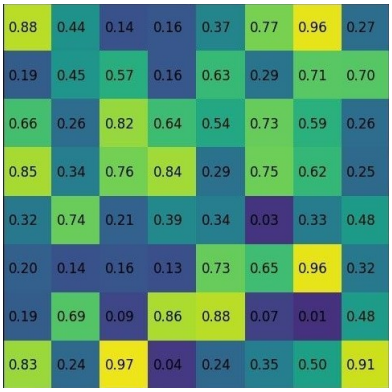


Figure 6.5: Input feature map.

Assume we also have a region suggestion (0, 3), (7, 8). It might be like this in the illustration in 6.6:

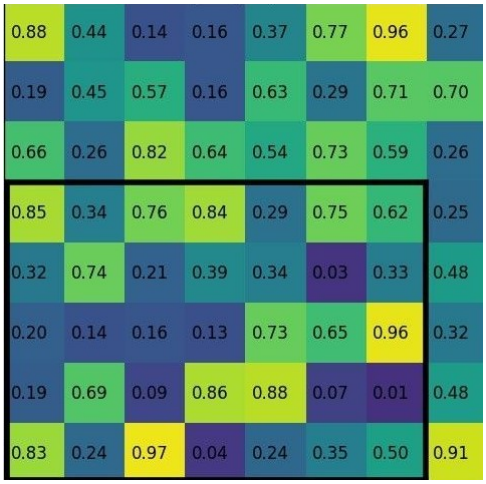


Figure 6.6: Region Proposal.

As the output size is 2x2, we'll divide it into 2x2 sections as shown in fig. 6.7.

The area of interest does not need to be exactly divided by the number of pooling sections.

The final output of the Region of Interest pooling will be as shown in fig. 6.9

The output of the ROI pooling layer is the fixed size feature maps corresponding to both human bounding boxes and object bounding boxes.

Now, after passing the instance level feature map for human and objects through the 5th convolutional block of the Residual Network and the Global Average Pooling Layer and then a fully connected layer, we get the appearance features f_h for human and f_o for object instances. We consider m as the number of human instances in the image and n as the number of object instances.

Now at the input of the LSTM based context module, we have appearance features for human f_h, appearance features for objects f_o, image level features, visual scanpaths and the stacked Human and Object Bounding boxes for each image.

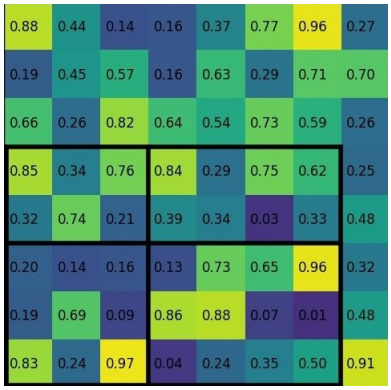


Figure 6.7: Region Proposal.

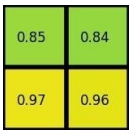


Figure 6.8: Maxpool.

This module is capable of learning the sequence in which each object in the scene is being observed. The output of this module is used for the score fusion and action classification.

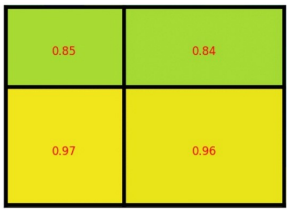


Figure 6.9: Output.

Let's look into the details of the LSTM based context module and score fusion process.

6.1.1 LSTM Based Context Module

In this section we'll discuss the detailed architecture of our proposed module.

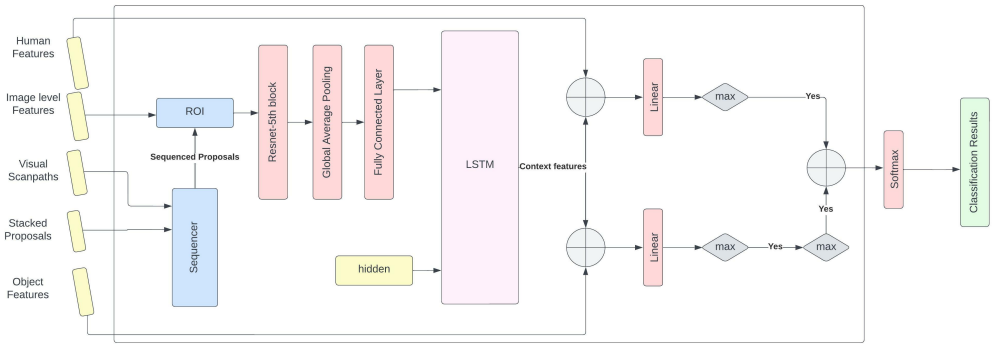


Figure 6.10: LSTM Based Context Module.

As shown in fig. 6.10, we have human features, object features, visual scanpaths and the human and object bounding boxes at the input of the context module.

Sequencer

A sequencer algorithm is developed to sequence all the object proposals in accordance with the visual scanpaths. For sequencing, Human bounding boxes and object bounding boxes are stacked together. Then the distance from each fixation point to the center of all the bounding boxes is calculated. Bounding box with the least distance is given most priority. In this way , all the bounding boxes are arranged according to the fixation points of visual scanpaths.

Sequenced proposals alongwith the image level features are fed to the Region of interest pooling layer. Output of the ROI pooling layer is the fixed size feature map which is again fed to the 5th block of the Residual Network. Next, the output of the Resnet block is fed to the Global average pooling layer and then it is passed to a fully connected layer.

LSTM

Output of the fully connected layer is a $(m + n) \times 1024$ dimensional appearance features (where m is the number of human instances and n is the number of object instances) which are passed to the LSTM. These features inherit the order of fixation points of the visual scanpaths which was provided by the sequencer algorithm. The order inherited in these features is learnt by the LSTM model.

Appearance features are fed to the LSTM in a sequence created using the human gaze on the image. But before feeding to the LSTM, positional encoding is added

with each of the instance appearance feature. For positional encoding, sine and cosine encodings are used as provided in the paper [11].

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{6.1}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{6.2}$$

where i is the dimension and pos is the location. That is, each positional encoding dimension corresponds to a sinusoid. From 2π to 10000π, the wavelengths form a geometric progression. We picked this function because we thought it would be simple for the model to learn to attend to relative locations, since PE_{pos+k} can be written as a linear function of PE_{pos} [11] for any fixed offset k.

Score Fusion

The LSTM's output now consists of context features generated for each of the eight observers' scanpaths. The LSTM's output is combined with human f_h and object f_o appearance features, and the resulting features are passed through a linear layer. As a result, the highest score from each of the eight observers' scanapath is used.

The intuition behind this model is that when the experiment on the SMI iView machine was conducted to collect the eye movement data of human observers, they gazed only at the specific areas of the image and traced the scene in a particular order for each image.

So, these key areas alongwith the sequence in which they are being observed could be learned by the an LSTM to give a particular output that can be assigned to a particular action category.

Figure 6.11: Image from VOC 2012 [27]

Suppose the images in the fig. 6.11 and fig. 6.12 are observed by the viewers. The viewer first observes the men and then observes the cycle. So, at the t^{th} time instant when viewer is observing the cycle, LSTM can learn the objects and their respective positions observed before t^{th} time instant. So, it can learn weather human

Figure 6.12: Image from VOC 2012 [27]

.

was at the top of the cycle or was just standing near the cycle and it can help the model recognize the action.

Chapter 7

Experimental Results

We employed the Pascal VOC 2012 dataset to test our model [27]. The dataset contains 9157 images that are divided into two groups: the training set, which contains 2296 images, and the validation set, which contains 2292 images. The rest is utilised for testing purposes.

We have used Residual networks with different depths and different hyperparameter settings. Let's look at some of the best generated results out of all the experiments we have done.

7.1 Resnet-50 As Feature Extractor

Learning Rate : 3×10^{-5}
No. of epochs : 15
Optimizer : Adam
Feature Extractor Network : Resnet-50

Methods	Jumping	Phoning	Playing Instrument	Reading	Riding Bike	Riding Horse	Running	Taking Photos	Using Computer	Walking	mAP
R*CNN	88.9	79.9	95.1	82.2	96.1	97.8	87.9	85.3	94.0	71.5	87.9
Attention	87.8	78.4	93.7	81.1	95.0	97.1	**96.0**	85.5	93.1	73.4	87.1
Part Action	**89.6**	86.9	94.4	**88.5**	94.9	97.9	91.3	87.5	92.4	76.4	90.0
HORN	89.2	**89.8**	96.5	87.6	**98.2**	**99.1**	92.3	**91.6**	95.2	**79.2**	**91.9**
Ours	88.06	86.41	**97.70**	87.01	96.72	98.97	88.62	89.75	**94.96**	77.11	90.53

Figure 7.1: Experiment 1: Results on Resnet-50.

When the Resnet-50 was used as the feature extractor, the mean average precision that we got is 90.53% on the validation dataset which is a little lesser than the state-of-the-art mAP in case of the Human-Object-Relation-Network. Looking at the individual classes, the average precision for "Playing Instrument" and "Using Computer" classes has surpassed all the previous works.

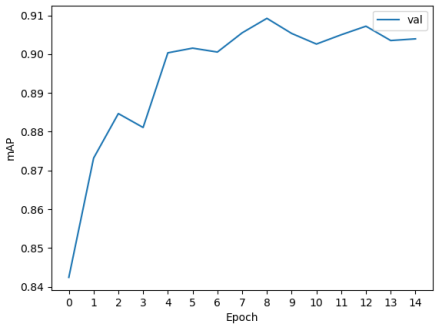


Figure 7.2: Experiment 1: mAP curve.

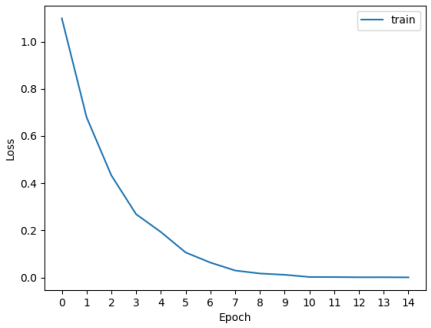


Figure 7.3: Experiment 1: Loss Curve.

The accuracy curve and the loss curve for this particular network are shown in fig. 7.2 and fig. 7.3.

7.2 Resnet-101 As Feature Extractor

Learning Rate : 3×10^{-5}
No. of epochs : 15
Optimizer : Adam
Feature Extractor Network : Resnet-101

With the Resnet-101, the mean average precision that we got is 91.65% on the validation dataset and it shows a little improvement in comparison to the Resnet-50 network. But still it is little lesser than the state-of-the-art mAP in case of the Human-Object-Relation-Network. Average Precision in case of the "Phoning", "Playing Instrument", "Taking Photos" and "Using Computer" has surpassed the Average Precision in respective classes for our model. Except the Human-Object-Relation-Network [1], mean average precision for our model is better than all other previous works.

The accuracy curve and the loss curve for this particular network are shown in fig. 7.5 and fig. 7.6. As the depth of Residual network is increased in this case,

Methods	Jumping	Phoning	Playing Instrument	Reading	Riding Bike	Riding Horse	Running	Taking Photos	Using Computer	Walking	mAP
R*CNN	88.9	79.9	95.1	82.2	96.1	97.8	87.9	85.3	94.0	71.5	87.9
Attention	87.8	78.4	93.7	81.1	95.0	97.1	**96.0**	85.5	93.1	73.4	87.1
Part Action	**89.6**	86.9	94.4	**88.5**	94.9	97.9	91.3	87.5	92.4	76.4	90.0
HORN	89.2	89.8	96.5	87.6	**98.2**	**99.1**	92.3	91.6	95.2	**79.2**	91.9
Ours	89.10	**90.29**	**96.82**	88.47	97.59	98.98	91.79	**91.91**	**95.29**	76.25	**91.65**

Figure 7.4: Experiment 2: Results on Resnet-101.

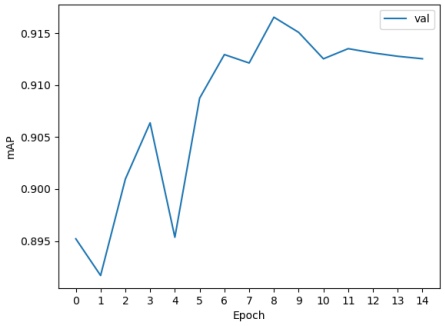


Figure 7.5: Experiment 2: mAP.

more fluctuations are observed in mAP curve.

7.3 Resnet-152 As Feature Extractor

Learning Rate : 3×10^{-5}
No. of epochs : 20
Optimizer : Adam
Feature Extractor Network : Resnet-152

With the Resnet-152 as the feature extractor, the mean average precision that we got is 92.00% on the validation dataset which is better than the mAP in case of both Resnet-50 and Resnet-101. Also, with this mAP, we have surpassed the state-of-the-art mAP in case of Human-Object-Relation-Network. Average Precision in case of the "Playing Instrument", "Reading", "Taking Photos" and "Walking" has surpassed the Average Precision in respective classes for our model.

The accuracy curve and the loss curve for this particular network are shown in fig. 7.8 and fig. 7.9. Even though the fluctuation were also unavoidable in this case, this model reached highest mean average precision compared to all the previous work.

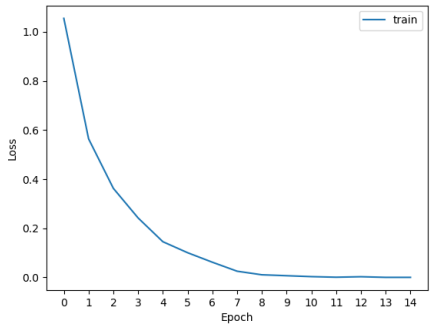


Figure 7.6: Experiment 2: Loss Curve.

Methods	Jumping	Phoning	Playing Instrument	Reading	Riding Bike	Riding Horse	Running	Taking Photos	Using Computer	Walking	mAP
R*CNN	88.9	79.9	95.1	82.2	96.1	97.8	87.9	85.3	94.0	71.5	87.9
Attention	87.8	78.4	93.7	81.1	95.0	97.1	**96.0**	85.5	93.1	73.4	87.1
Part Action	**89.6**	86.9	94.4	88.5	94.9	97.9	91.3	87.5	92.4	76.4	90.0
HORN	89.2	**89.8**	96.5	87.6	**98.2**	**99.1**	92.3	91.6	**95.2**	79.2	91.9
Ours	89.14	89.12	**96.87**	**90.02**	97.78	98.72	92.15	**91.85**	94.62	**79.77**	**92.00**

Figure 7.7: Experiment 3: Results on Resnet-152.

7.4 Addition of Scene-Object Interaction Block

In this experiment, we also considered the scene-object interaction. For this, direct output of the Residual Network (bypassing the ROI pooling) is fed to the Global Average Pooling layer. The output of the GAP layer is the scene feature and is added to the object appearance features. These scene-object interaction features were passed to the LSTM-based context module in place of the object appearance features and rest of the computation remains same.

Learning Rate : 3×10^{-5}
No. of epochs : 20
Optimizer : Adam
Feature Extractor Network : Resnet-152

After adding the scene-object interaction features, the mean average precision that we got is 92.62% on the validation dataset which is best mAP among the above all the results generated. This mAP is also greater than the state-of-the-art mAP in case of Human-Object-Relation-Network. Except for the "Riding Horse" and "Running" classes, average precision for all the other classes is greater in this case.

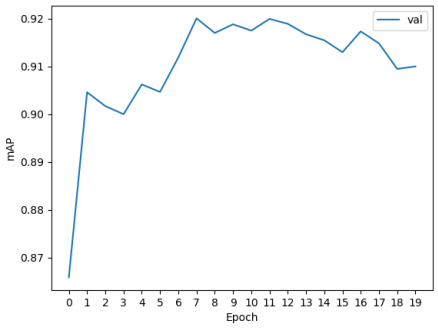


Figure 7.8: Experiment 3: mAP.

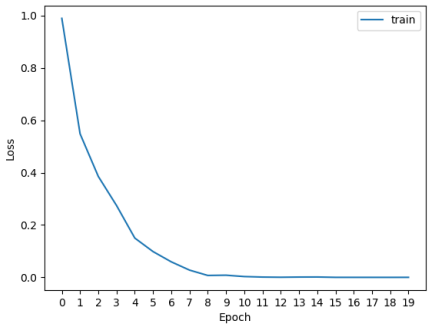


Figure 7.9: Experiment 3: Loss Curve.

7.5 Freezing the Sequencer Algorithm

In this experiment, we removed the sequencer algorithm from our model to check the significance of sequencing the object proposals in the order of fixation points of visual scanpaths.

Learning Rate : 3×10^{-5}
No. of epochs : 20
Optimizer : Adam
Feature Extractor Network : Resnet-152

The mean average precision in this case reduced a little as compared to the experiment 4. So, we can conclude that sequencing the object proposals in the order of the fixation points of the visual scanpaths has significant impact on the model and our assumption that human traces an image in particular order to recognize an action from an image, is true.

The accuracy curve and the loss curve for this particular network are shown in fig. 7.11 and fig. **??** respectively.

Methods	Jumping	Phoning	Playing Instrument	Reading	Riding Bike	Riding Horse	Running	Taking Photos	Using Computer	Walking	mAP
R*CNN	88.9	79.9	95.1	82.2	96.1	97.8	87.9	85.3	94.0	71.5	87.9
Attention	87.8	78.4	93.7	81.1	95.0	97.1	**96.0**	85.5	93.1	73.4	87.1
Part Action	89.6	86.9	94.4	88.5	94.9	97.9	91.3	87.5	92.4	76.4	90.0
HORN	89.2	89.8	96.5	87.6	98.2	**99.1**	92.3	91.6	95.2	79.2	91.9
Ours	**90.21**	**91.03**	**97.60**	**89.83**	**98.40**	99.06	92.44	**92.42**	**95.72**	**79.63**	**92.63**

Figure 7.10: Experiment 4: Results after scene-object interaction.

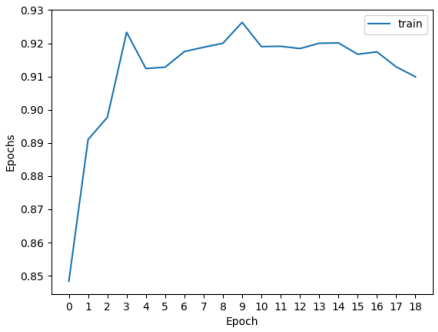


Figure 7.11: Experiment 4: mAP curve.

Methods	Jumping	Phoning	Playing Instrument	Reading	Riding Bike	Riding Horse	Running	Taking Photos	Using Computer	Walking	mAP
R*CNN	88.9	79.9	95.1	82.2	96.1	97.8	87.9	85.3	94.0	71.5	87.9
Attention	87.8	78.4	93.7	81.1	95.0	97.1	**96.0**	85.5	93.1	73.4	87.1
Part Action	**89.6**	86.9	94.4	88.5	94.9	97.9	91.3	87.5	92.4	76.4	90.0
HORN	89.2	89.8	96.5	87.6	**98.2**	**99.1**	92.3	**91.6**	95.2	**79.2**	91.9
Ours	88.59	**91.57**	**96.67**	**89.78**	98.07	98.76	91.48	91.48	**95.12**	78.19	**91.97**

Figure 7.12: Experiment 5: Results after removing Sequencer Algorithm.

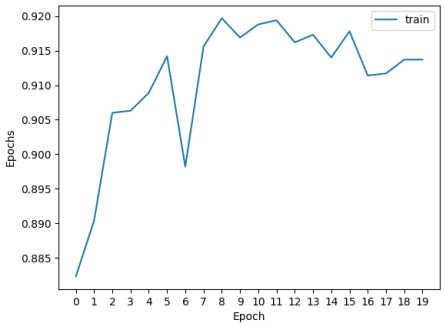


Figure 7.13: Experiment 5: mAP curve.

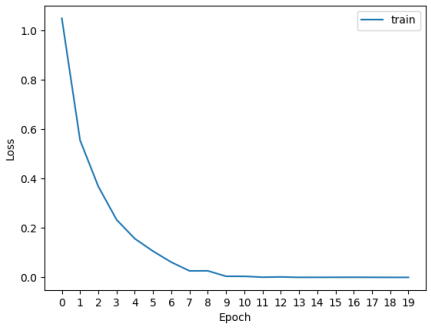


Figure 7.14: Experiment 5: Loss curve.

Chapter 8

Future Work

Having done a number of experiments on PASCAL VOC 2012 dataset on our proposed model, evaluation of the model on test dataset is still remaining. Other than this, proposed model is not generalized to load any other dataset than the PASCAL VOC 2012. So, a dataloader part can be written for other datasets.

Object bounding boxes used in this thesis were generated using the Faster-RCNN method. Other methods like Mask-RCNN can be explored to find more accurate bounding boxes.

Chapter 9

Conclusion

In this thesis, we performed a detailed literature review, understood the nuances of Action Recognition in Still Images. We looked at the several approaches towards solving the Action Recognition problem. We have also implemented a new model which is based upon an LSTM based context module. This architecture is designed considering the human visual mechanism. We have used Resnet models with different depths to extract the features. With increased depth, the mean average precision has increased. With Resnet-152, we got a mAP of 92.00% and thus our model has shown better results than the Human-Object-Relation-Network and Region based Convolutional Neural Network.

References

[1] Wentao Ma and Shuang Liang. Human-Object Relation Network For Action Recognition In Still Images, 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020.

[2] Lei Wang et. al. Human Action Recognition by Learning Spatio-Temporal Features With Deep Neural Networks, 2018 IEEE.

[3] Gary Ge. et. al. Action Classification in Still Images Using Human Eye Movements Computer Vision and Pattern Recognition 2015

[4] Georgia Gkioxari. et. al. Contextual Action Recognition with R*CNN International Conference on Computer Vision

[5] Yan, Shiyang and Smith, Jeremy S. and Lu, Wenjin and Zhang, Bailing, Multibranch Attention Networks for Action Recognition in Still Images, IEEE Transactions on Cognitive and Developmental Systems, 2018

[6] Wu, Wei and Yu, Jiale. An Improved Deep Relation Network for Action Recognition in Still Images. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

[7] Daria Stefic, Ioannis Patras. Action recognition using saliency learned from recorded human gaze Image and Vision Computing, Volume 52, 2016, Pages 195-205, ISSN 0262-8856, https://doi.org/10.1016/j.imavis.2016.06.006. (https://www.sciencedirect.com/science/article/pii/S026288561630107X)

[8] Delaitre, Vincent Sivic, Josef and Laptev, Ivan. Learning person-object interactions for action recognition in still images, 2011

[9] Fathi, Alireza and Li, Yin and Rehg, James M. Learning to Recognize Daily Actions Using Gaze Computer Vision – ECCV 2012

[10] Liu, Lu and Tan, Robby T. and You, Shaodi. Loss Guided Activation for Action Recognition in Still Images https://arxiv.org/abs/1812.04194 arXiv.org perpetual, non-exclusive license

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need Advances in neural information processing systems, 2017, pp. 5998–6008.

[12] D. Girish, V. Singh and A. Ralescu, "Understanding action recognition in still images," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1523-1529, doi: 10.1109/CVPRW50498.2020.00193.

[13] Mallya, A., Lazebnik, S. Learning models for actions and person-object interactions with transfer to question answering in European Conference on Computer Vision. pp. 414–428. Springer (2016)

[14] Zhao, Z., Ma, H., You, S. Single image action recognition using semantic body part actions. in 2017 IEEE International Conference on Computer Vision (ICCV), Venice. pp. 3411–3419 (2017)

[15] Oquab, M., Bottou, L., Laptev, I., Sivic, J. Learning Learning and transferring mid-level image representations using convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. pp. 1717–1724. IEEE (2014)

[16] Zhang, Y., Cheng, L., Wu, J., Cai, J., Do, M.N., Lu, J. Action recognition in still images with minimum annotation efforts. IEEE Transactions on Image Processing 25(11), 5479–5490 (2016)

[17] Ranjan, R., Patel, V.M., Chellappa, R. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)

[18] Simonyan, K., Zisserman, A. Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems. pp. 568– 576 (2014)

[19] Girdhar, R., Ramanan, D. : Attentional pooling for action recognition. In: NIPS (2017)

[20] P. Li, J. Ma, and S. Gao, "Actions in still web images: visualization, detection and retrieval," in International Conference on Web-Age Information Management. Springer, 2011, pp. 302–313.

[21] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 3, pp. 601–614, 2011.

[22] F. Sener, C. Bas, and N. Ikizler-Cinbis, "On recognizing actions in still images via multiple features," in European Conference on Computer Vision. Springer, 2012, pp. 263–272.

[23] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in Computer Vision and Pattern Recognition, CVPR, 2011. IEEE, 2011, pp. 3177–3184.

[24] Samitha Herath, Mehrtash Harandi, Fatih Porikli. Going deeper into action recognition: A survey, Image and Vision Computing, Volume 60, 2017, Pages 4-21, ISSN 0262-8856, https://doi.org/10.1016/j.imavis.2017.01.010. (https://www.sciencedirect.com/science/article/pii/S0262885617300343)

[25] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 1440–1448

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[27] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge," International journal of computer vision, vol. 88, no. 2, pp. 303–338, 2010.

[28] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, Mu Li. "Bag of Tricks for Image Classification with Convolutional Neural Networks". 2018, arXiv.org perpetual, non-exclusive license