

Video Object Detection and Identification in Dynamic Environment

by

Mahir Manishbhai Shah
202011002

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY



June, 2022

Declaration

I hereby declare that

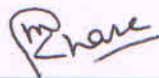
- i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,
- ii) due acknowledgment has been made in the text to all the reference material used.



Mahir Manishbhai Shah

Certificate

This is to certify that the thesis work entitled "Object Identification and Segmentation in Dynamic Environment" has been carried out by **Mahir Manishbhai Shah (202011002)** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under our supervision.



Dr. Manish Khare
Thesis Supervisor



Dr. Ahlad Kumar
Thesis Co-Supervisor

Acknowledgments

I would like to thank my thesis supervisor Dr. Manish Khare and my thesis co-supervisor, Dr. Ahlad Kumar, for their continuous support and guidance throughout my journey and for keeping patience and believing in me even though I had no background in the Image Processing Domain. I am greatly honored to work with them. Their friendly nature, willingness to help whenever I needed help, suggestions for improving my thesis work, and excellent domain knowledge have helped me to bring out the best in me which would have been otherwise difficult for me without their guidance.

I would also like to thank all my friends, especially to Abhishek, and Krunal for always helping me whenever I needed their help. I would also like to thank Kashyap for always giving me the moral support to give my best for my thesis work.

I would also like to thank the institute and the help desk team for providing me with the necessary software and hardware support for carrying out my thesis work.

Lastly, I would like to thank my parents and my family members, who always believed in me and supported me. Thank you for giving your blessings and moral support.

Mahir. M. Shah.

Contents

Abstract	v
List of Principal Symbols and Acronyms	vi
List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 General Introduction	1
1.2 Thesis Objectives and Problem Description	2
1.3 Proposed Work	3
1.4 Contributions	3
1.5 Thesis Organization	4
2 Related Works and Literature Survey	5
2.1 Object Identification in Still Images	5
2.2 Object detection and identification in dynamic environment	6
2.3 Literature Survey	7
2.3.1 You only look once(YOLO)[1]	7
2.3.2 Flow-Guided-Feature-Aggregation(FGFA)[2]	8
2.3.3 Relation Distillation Networks for Video Object Detection (RDN) [3]	9
2.3.4 Sequence Level Semantics Aggregation for Video Object De- tection (SELSA) [4]	10
2.3.5 Fully Motion-Aware Network for Video Object Detection (MANet) [5]	11
3 Proposed Methodology	13
3.1 Model Architecture	13
3.2 Feature Warping	17

3.2.1	Feature Warping for Initialising weights	17
3.2.2	Feature Warping for trained weights	17
3.3	Feature Aggregation	18
3.3.1	Feature Aggregation for Initialising weights	18
3.3.2	Feature Aggregation for trained weights	18
3.4	Loss Function	19
3.4.1	Loss Function for First Model	19
3.4.2	Loss Function for Second Model	19
4	Experimental setup	20
4.1	ImageNet VID 2015 Dataset[6]	20
4.2	YouTube-8M Dataset[7]	20
4.3	Different types of motions in videos	21
5	Implementation Details	22
5.1	Implementation Details of First Model	22
5.1.1	Feature network	22
5.1.2	Detection network	22
5.1.3	Training	23
5.2	Implementation Details of Second Model	23
5.2.1	Feature Network	23
5.2.2	Detection Network	23
5.2.3	Training	24
6	Expiremental Results and Analysis	25
6.1	ImageNet VID 2015 Dataset	25
6.2	YouTube-8M Dataset	30
7	Conclusions and Future Works	36
7.1	Conclusion	36
7.2	Future Works	36
	References	37

Abstract

Object Detection and Identification in the field of computer vision is widely regarded as one of the most difficult problems in computer science. Yet it is one of the most rising topics in recent years due to the advancement of the computer hardware technologies like GPUs. The task of Object Detection and Identification can be further divided into two categories:

1. Object Detection and Identification in still images.
2. Object Detection and Identification in dynamic environments.

Due to the advancements in computer hardware like GPUs, deep neural network based methods have shown great accuracy and most of the state-of-the-art methods for still images are based on deep neural networks. Extending these state-of-the-art object detectors for still images into dynamic environments is not easy as we see a drop in accuracy because of the deteriorated object appearances like rare poses, motion blurs, video defocus, and part or full occlusion. The reason for the decrease in accuracy is that still image detectors do not take into account the temporal information contained in videos when detecting the objects in dynamic environment like videos. To improve the accuracy of the state-of-the-art detectors in the dynamic environment like videos, different methods have been developed which takes into consideration temporal information present in videos.

In this thesis work, we have tried to increase the accuracy of state-of-the-art object detectors by trying to use the knowledge of the previously trained model as a reference to another model. In our work, we have also tried to simplify the architecture when we combine two different models without incurring a loss in the accuracy. In our thesis work, the first model that we have trained is an pixel-level method and the second model that we have trained is an instance-level method. We have tested our approach on the ImageNet VID dataset and YouTube-8M dataset. Results show that our approach has obtained improved results in instance-level object detection methods.

List of Principal Symbols and Acronyms

CNN	Convolutional Neural Network
DFE	Deep Feature Flow
FC	Fully Connected
FGFA	Flow-Guided Feature Aggregation for Video Object Detection
IoU	Intersection over Union
LCFN	Local Context Feature Network
MANet	Fully Motion-Aware Network for Video Object Detection
mAP	Mean Average Precision
R-CNN	Regions with Convolutional Neural Networks
ROI	Region of Interest
RPN	Regional Proposal Network
SELSA	Sequence Level Semantics Aggregation for Video Object Detection
SGD	Stochastic Gradient Descent
YOLO	You Only Look Once

List of Tables

6.1	mAP(%) values of all 30 categories for ImageNet VID 2015 Dataset	27
6.2	Accuracy Table for ImageNet VID 2015 Dataset	28
6.3	mAP(%) values of all 30 categories for YouTube-8M Dataset	33
6.4	Accuracy Table for YouTube-8M Dataset	35

List of Figures

1.1	Deteriorated Object appearance in videos	2
2.1	Architecture of YOLO	7
2.2	Architecture of FGFA	8
2.3	Architecture of RDN	9
2.4	Architecture of SELSA	10
2.5	Architecture of MANet	11
3.1	Architecture of Proposed Work	14
3.2	Comparison of accuracy for partial occlusion	16
4.1	Video Snippets of objects having different motion	21
6.1	Visualization of results for ImageNet VID 2015 Dataset	26
6.2	Confusion matrix of 30 categories for ImageNet VID 2015 Dataset .	29
6.3	Accuracy comparison of proposed method result with other state-of-the-art methods for ImageNet VID 2015 Dataset	30
6.4	Visualization of results for YouTube-8M Dataset	32
6.5	Confusion matrix of 30 categories for YouTube-8M Dataset	34
6.6	Accuracy comparison of proposed method result with other state-of-the-art methods for YouTube-8M Dataset	35

CHAPTER 1

Introduction

Object Detection and Identification in images is an effort made by the researchers, and computer scientists to learn more about how humans detect and identify objects not only in still images but also in videos and then apply this knowledge to improve the accuracy of the human created models to detect and identify objects in images as well as in dynamic environments like videos. This chapter firstly discusses about Object detection and Identification, then it talks about challenges in the area of object Identification in dynamic environment, objective of the thesis and, problem description, followed by the proposed work and the contributions made by this thesis work and, at last ends with thesis organization.

1.1 General Introduction

In recent years, the world of computer vision has seen and witnessed significant progress in the domain of object detection[8], especially in the field of video object detection. The advancement in the deep convolution networks and hardware components like GPUs have contributed tremendously in getting better accuracy and improving the state-of-the-art object detectors not only for still images but also for video object detection. state-of-the-art object detectors for still images mostly share a similar two-stage detection network architecture structure. In the first stage, deep convolution neural networks are applied on the images, and a set of features are generated from them[9, 10, 11, 12], In the second stage, a shallow detection-specific network is applied to generate detection results from the input feature maps of the first stage[13, 14, 15, 16].

1.2 Thesis Objectives and Problem Description

Deep convolution networks have achieved great accuracy in still images but when these state-of-the-art object detectors for still images are applied directly to the videos for object detection it becomes a bit challenging. Loss in object detection accuracy is seen in video object detection because of the deteriorated object appearances such as motion blur, rare poses, video defocuses, occlusion, etc. as shown in Figure 1.1 which is anyways not observed in still images.



Figure 1.1: Deteriorated Object appearance in videos[2].

Most of the state-of-the-art object detectors for still images have failed in giving good accuracy when applied directly to the video object detection due to the deteriorated object appearances as shown in Figure 1.1. These state-of-the-art object detectors for still images don't make use of the rich temporal information which is present in videos. Many new state-of-the-art techniques have been developed which take the temporal information into account in which some methods work at the pixel level whereas some methods work at the semantic level to achieve better accuracy. Many methods/techniques even try to go one step further by combining both methods to improve accuracy which in turn makes the network architecture complex.

Video object detection and identification even though being a difficult task has gained a lot of importance in the past few years because of its applications in the real world. For example, vehicle number plate detection of moving vehicles,

person detection in automated CCTV surveillance, or taking the example of the recent pandemic situation, where wearing a mask was made mandatory by the government and failing to comply with the rule resulted in a fine. But the government found it difficult to collect the fine when people gathered in numbers in a public place, where identifying the mask using CCTV live feed can become very helpful. The application of object identification is increasing as the real world is a dynamic world unlike a still image having static background.

1.3 Proposed Work

In our thesis work, we try to look deeper into the video object detection domains model and try to simplify the network architecture when the pixel level and instance level techniques are combined without the loss of accuracy. In this work, we try to incorporate the use of previously trained weights of pixel-level network architecture[2] as a reference for the semantic level architecture[4] for generating feature maps and detection results.

We propose to use of transfer learning in the domain of video object detection where one model is based on pixel-level method[2] and another model is based on instance-level method[4] to achieve excellent accuracy with a simplified network architecture model. We first train our model via the pixel-level method FGFA[2] and then use its trained weights to train an instance-level model architecture SELSA[4]. We have trained both of our models on the ImageNet VID 2015 dataset[6] which is considered to be a benchmark dataset for video object detection. We have tested our approach on ImageNet VID 2015 dataset[6] and the YouTube-8M dataset [7].

1.4 Contributions

The thesis contributions are as follows:

- In our thesis work, we have introduced the use of transfer learning approach to fully utilize the video information by using the weights of the trained model that is a pixel-level based method[2] as a reference for another model that is an instance-level based method [4].
- We have tested our proposed approach on the ImageNet VID 2015 dataset[6] and the YouTube-8M dataset [7] which are large-scale datasets for the task

of video object detection and identification in dynamic environment and demonstrated improvement over previous methods.

1.5 Thesis Organization

The Thesis report is further divided as follows: Chapter 2 talks about the related works that have been conducted in the domain of video object detection and identification along with the literature survey. Chapter 3 talks about our proposed methodology and model architecture of the proposed work. Chapters 4 and 5 discuss about the experimental setup, the datasets that we have used for experimentation, and the implementation details of both the models that we have used along with the hardware details. Chapter 6 discusses about the results of our proposed approach on the ImageNet VID 2015 dataset[6] and YouTube-8M dataset [7], along with the comparisons with state-of-the-art methods namely FGFA[2], SELSA[4], and MANet[5]. Finally, Chapter 7 discusses about the conclusions and future work for the conducted studies.

CHAPTER 2

Related Works and Literature Survey

Object detection in videos refer to the process of locating objects in video sequence's frame. Object detection mechanism is required for every identification process, whether in every frame or when an object appears for the first time in a frame.. Most object detection mechanism uses information from a single frame for detecting an object. But some of the object detection mechanisms used temporal information which is computed from a sequence of nearby frames. There are several object detection mechanisms that have been developed for video object detection and identification task. The first step of real-time object detection and identification is to identify the region of interest in the video. This chapter discusses about the previous works that have been developed, in the domain of object detection and identification.

2.1 Object Identification in Still Images

Because of the development in the deep convolution neural networks, many state-of-the-art object detectors for still images are based on deep CNN[17, 18, 19, 20]. This deep CNN are divided into two types. The first one is a single-stage detector and the second one is a two-stage detector. Two-stage detectors typically generate the detected output in two stages. For example R-CNN[8] in the first stage will extract regional features from a backbone network that is made up of Deep convolution neural networks, and in the second stage uses the regions of the first stage for classification and generating bounding boxes. Another two-stage detector Fast R-CNN[21] introduced the ROI Pooling operation for speeding up the extraction of the regional feature in stage one of R-CNN, selective search[22] is used for generating region proposals. Faster R-CNN[23] proposed RPN to generate Region Proposals, using the Fast R-CNN backbone. Another method R-FCN[24] introduced ROI Pooling operation which is position sensitive to improve the detection efficiency. Another recent work is CCD-Net[25], It uses an LCFN module for ex-

tracting features of neighboring objects. To focus more on valuable features, Hybrid Attention Pyramid Network(HAPN) is a module that is deployed by CCD-Net. Paper MSFYOLO[20] proposes an object detection algorithm to detect small objects via multi-scale feature fusion. The proposed multi-scale feature learning technique combines concrete and abstract characteristics by learning shallow features at a shallow level and deep features at a deep level. Based on the multi-scale feature learning network, it produces a feature pyramid for object detection by combining global and local information.

One stage detectors instead of having two stages have only one stage, which performs the task of generating feature maps as well as bounding boxes for the objects. One-stage detectors are usually faster than the two-stage detectors because of not have to do extra work. You only look once(YOLO)[1] and its variations YOLY9000[26] and YOLOv3[27], SSD[14], DSSD[28], and Lin -et al[23] are some one stage detectors.

2.2 Object detection and identification in dynamic environment

ImageNet was among the very first to introduce a challenge for video object detection. Mostly all the methods made the use of temporal information via the “bounding box post-processing” step onto the final stage. T-CNN[14] instead of using bounding boxes at the final stage propagates them to the neighboring nearby frames using optical flow precomputed beforehand and then applies the tracking algorithms from high confidence bounding boxes to generate tubelets. Both the boxes and tubelets based on tubelets classification are re-scored. Seq-NMS[29] from consecutive frames constructs sequences on the nearby neighboring bounding boxes having a high-confidence score. The post-Processing step is formulated as a multi-object tracking problem in MCMOT[30]. To know whether the bounding boxes are associated with the tracked item or not, many hand-craft rules like color or motion clues, detector confidences, etc are used which can be further used to refine the results of the tracking. DFF[31] is among the first work to adopt fine tuned optical flow computation in-network. It uses the optical flow generated from the FlowNet[31] for propagating and aligning the selected keyframes features to the surrounding non-keyframes, minimising the redundant calculations and thus increasing the system speed. FGFA[2] is built on DFF[31] with an objective to improve the accuracy by aggregating the features using the optical flows on the keyframes. MANet[5] based on FGFA[2] and DFF[31] in ad-

dition to the pixel level, provides an instance-level feature calibration and aggregation module in FGFA[2] and then integrated both the levels via a motion pattern reasoning module. RDN[3] tries to learn the relation between the candidate boxes of nearby neighboring frames and use it to enhance the box-level features. MEGA[32] is also a method that tries to improve the accuracy by taking into consideration the local as well the global information to enhance the feature maps of every frame.

2.3 Literature Survey

2.3.1 You only look once(YOLO)[1]

YOLO tries to detect objects in images and videos by adopting the idea of how humans try to detect objects i.e. by taking into consideration the entire image or the entire frame of the video. YOLO is a one-stage detector that tries to both predict the bounding boxes and classify them together in one pass only. As a one-stage detector, YOLO has numerous advantages. By treating the object classification problem as a regression problem, YOLO is able to detect and classify objects exceptionally quickly. YOLO is fast enough to recognize objects in videos shot at 45-frame-per-second speed. Another faster version of YOLO based on Faster R-CNN has been developed, which can recognize objects in videos captured at the speed of 150 frames per second.

The second benefit that YOLO has over two-stage detectors is higher accuracy for detecting generalized representation of objects as YOLO looks at the entire image to detect and identify the objects. The architecture of YOLO is shown in Figure 2.1.

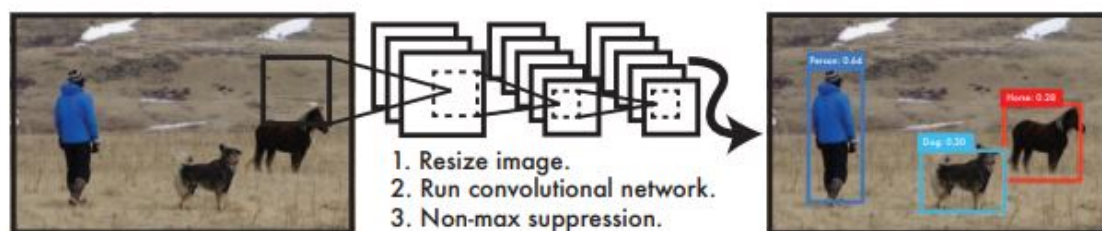


Figure 2.1: Architecture of YOLO[1].

YOLO for detection purpose first resizes the entire image and then divides the entire image into a matrix of $S \times S$ grids for detection. Each grid cell in the image predicts the B number of bounding boxes and the confidence score for the

predicted boxes once the CNN network is run on the entire image as shown in Figure 2.1. The bounding box having the center of the object is responsible for detecting the object of the image. YOLO being a one-stage detector detects objects faster than two-stage detectors and thus, can be used to detect objects in real-time.

2.3.2 Flow-Guided-Feature-Aggregation(FGFA)[2]

One of the reasons for object detectors of still images not performing very well in dynamic environments is because object detectors do not take into consideration temporal information present in the video to detect and identify objects. FGFA[2] is a method that runs on top of the object detectors and tries to use the temporal information of the video to improve the accuracy of the object detectors that consider the nearby neighboring frames. FGFA[2] takes into consideration previous the 9 frames and the 9 frames ahead of the current reference time as nearby neighboring frames for getting high accuracy. The architecture of FGFA is shown in Figure 2.2

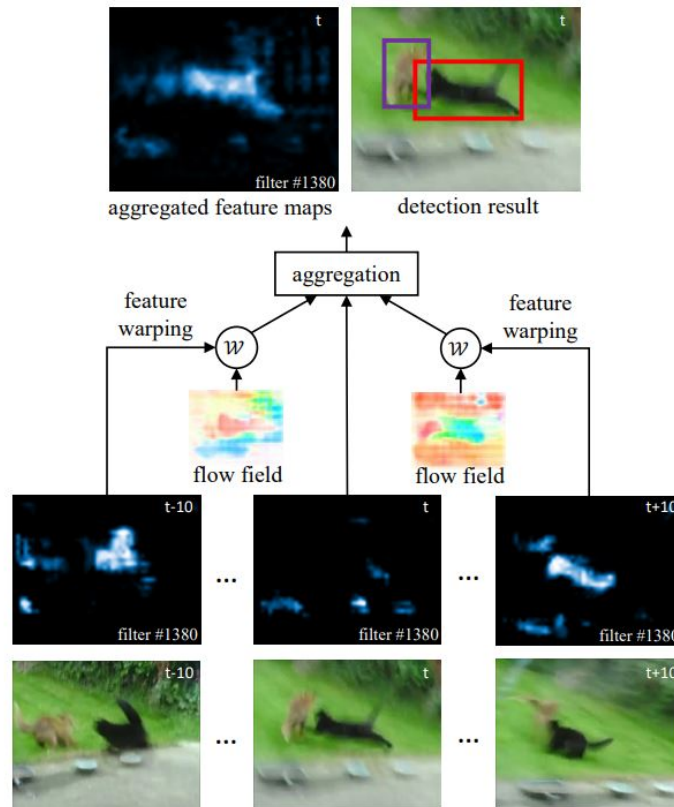


Figure 2.2: Architecture of FGFA[2].

FGFA[2] detects and identifies the objects in two stages. Firstly, FGFA[2] extracts the feature maps of the reference frames and the nearby neighboring frames.

These extracted feature maps are then warped to the reference frame, resulting in the reference frame having diverse information. Then in the second stage, it aggregates these warped features into the reference frame by giving higher priority to the feature maps of the near frames and less priority to the far away frames. After the aggregation is performed, these enhanced feature maps are then sent into the detection network for identifying the objects that are present in the current reference frame as shown in Figure 2.2.

2.3.3 Relation Distillation Networks for Video Object Detection (RDN) [3]

RDN is a novel approach that tries to improve the accuracy for object detection, that takes into consideration the relation of the same object between the different frames. RDN works in two stages: 1) Basic stage, and 2) Advance stage. The architecture of RDN is shown in Figure2.3.

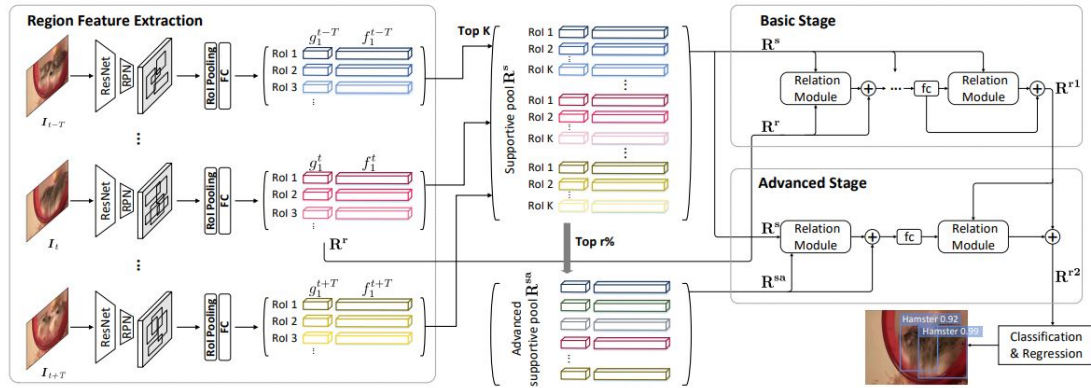


Figure 2.3: Architecture of RDN[3].

In the basic stage, RPN is employed to produce the object proposals(ROI) from the reference frames as well as the nearby neighboring frames. From the object proposals generated, top K proposals are selected and then sent into the relation module which aggregates the proposals of the nearby frames with the reference frame.

In the Advance stage, the top r% proposals of the top K proposals are used to aggregate the proposals of the frames with only high similarity with the objects in the reference frame. Advance stage is introduced in RDN to dilute the weightage of the object proposals that are not very important, as the basic stage tries to take into consideration all the proposals from the nearby top K frames.

Once the proposal from the advance stage is obtained, these proposals are aggregated with the proposals of the basic stage, and then these enhanced feature

proposals are used to detect and identify the objects from the current reference frames.

2.3.4 Sequence Level Semantics Aggregation for Video Object Detection (SELSA) [4]

To increase the accuracy of the video object detection task, SELSA[4] is a novel method that relies on the semantic level for object identification rather than depending on the optical flow of the pixels. SELSA[4] takes into consideration a group of pixels for identifying objects rather than working on each and every raw pixel of the input reference frame.

SELSA[4] works by aggregating the global level information into the feature proposals of the current reference frame. Methods that are based on optical flow can sometimes be inefficient as they do not try to look further beyond a certain threshold of frames that is the nearby neighboring frames. SELSA[4] tries to enhance the feature maps of the current reference frame by taking into account the global information and aggregating the feature maps having high similarity values. SELSA[4] looks one step further to enhance feature maps by clustering the feature maps having high similarity values. SELSA[4] first generates the proposals by running a proposal extraction network similar to the one which is used in pixel-level method. After the proposals are generated, proposals are warped using similarity between them which is calculated using cosine similarity. Proposals that have been warped are then aggregated using the SELSA[4] module and at last, a detection network is run over these aggregated proposals to identify the objects present in the frame as shown in Figure 2.4.

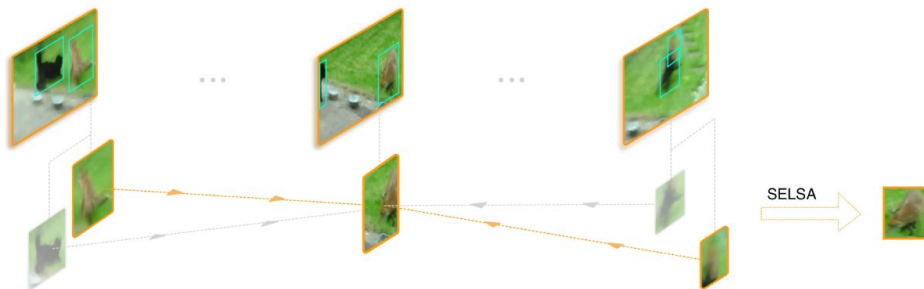


Figure 2.4: Architecture of SELSA[4].

2.3.5 Fully Motion-Aware Network for Video Object Detection (MANet) [5]

MANet[5] is a method that trains two methods first one being a pixel level method and the other one being an instance-level method and then combines their results at the runtime during the task of object detection and identification. Feature maps for the objects present in the current reference frame are generated by both methods, but feature maps of only one method are used for object detection and identification at runtime. MANet[5] has 3 modules in it, the first one is a pixel-level method, the second module is an instance-level method and the third one is a motion pattern based combination as shown in Figure 2.5.

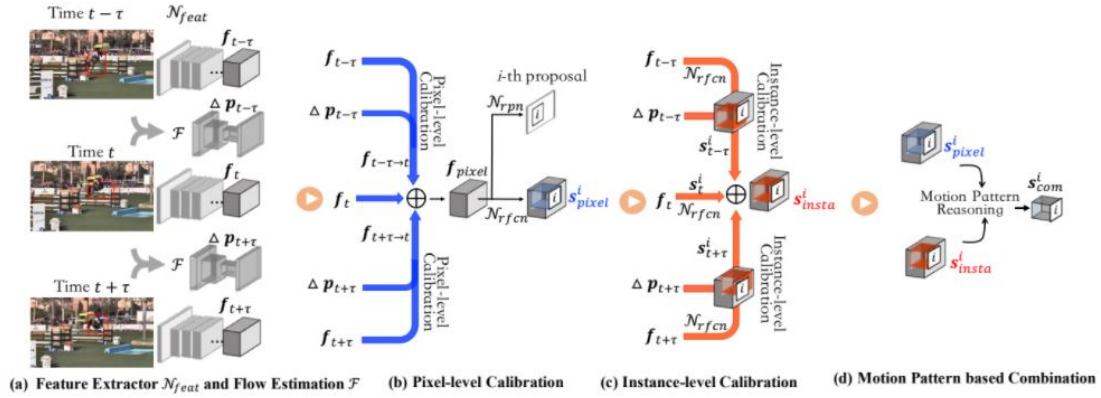


Figure 2.5: Architecture of MANet[5].

The First module in the MANet[5] first aggregates the feature maps of the nearby neighboring frames 9 frames(that is previous 9 frames from the current reference frame and 9 frames that are on time occurring ahead of the reference frame) into the feature maps of the current reference frame to enhance its feature maps. These feature maps from the output of the first model are stored and forwarded to the instance-level method and the motion pattern reasoning module.

The second instance-level module already gets the feature maps that are rich in information but they have local information which is not more than nearby 9 frames. The Instance-level module further enhances the feature maps of the aggregated feature maps by taking into consideration the global information of the nearby neighboring frames with a random set of indices. These enhanced aggregated feature maps are then forwarded to the motion pattern reasoning module as shown in Figure 2.5.

The Third module motion pattern reasoning module first finds the average of the (x_i/y_i) of all the bounding boxes from the feature maps that have been gen-

erated from both the methods. After that, if the value comes to be smaller then it means the object that is present is a rigid body and it will be beneficial to detect the object based on the instance-level method as the instance-level method is more robust toward change in the appearance of the objects. Similarly, if the value comes out more than the threshold value then the object is a non-rigid body that can be detected and identified better by the pixel level method as the pixel level method takes into account raw pixels directly. The motion pattern reasoning module's work is to decide the feature maps of which module should be used at the runtime to detect and identify the object given the input video frame.

CHAPTER 3

Proposed Methodology

In this work, our aim is to identify the objects for all the input video frames I_i , where $i = 1, 2, 3, \dots, N$, and output their respective class labels with high accuracy. In our work, we try to use the knowledge of a previously trained model for reference and tuning the previously trained weights in new architecture to get better accuracy, rather than training the model entirely from the scratch in less time and with fewer resources.

3.1 Model Architecture

Videos have rich temporal information which can be exploited and used to improve the accuracy of detecting and identifying the objects.

Our model architecture consists of a total of two models that we have used to get the improvement in accuracy of detecting and identifying the objects that are present in the input video frame of the video. The first model that we have used in our model architecture is a pixel-level based method[2], and the second model which is the final model in our model architecture is an instance-level based method[4].

For the first model's training purpose, optical flow is used to calculate the motion between the same object but in different frames of the input video sequence. The model has 5 blocks of convolution layer which are used to extract the feature maps from the input video frames where each layer is followed by an activation layer that activates neurons of the next layer based on the learned threshold values. Initially, threshold value of the activation layer is initialized randomly which is then relearned based on SGD training as the backbone which we are using is an R-CNN based backbone for object detection and identification in an dynamic environment. The activation layer is used after the convolution block to reduce the number of neurons in the next blocks to reduce the number of computations that are taking place. The architecture of our proposed model is shown in Figure 3.1.

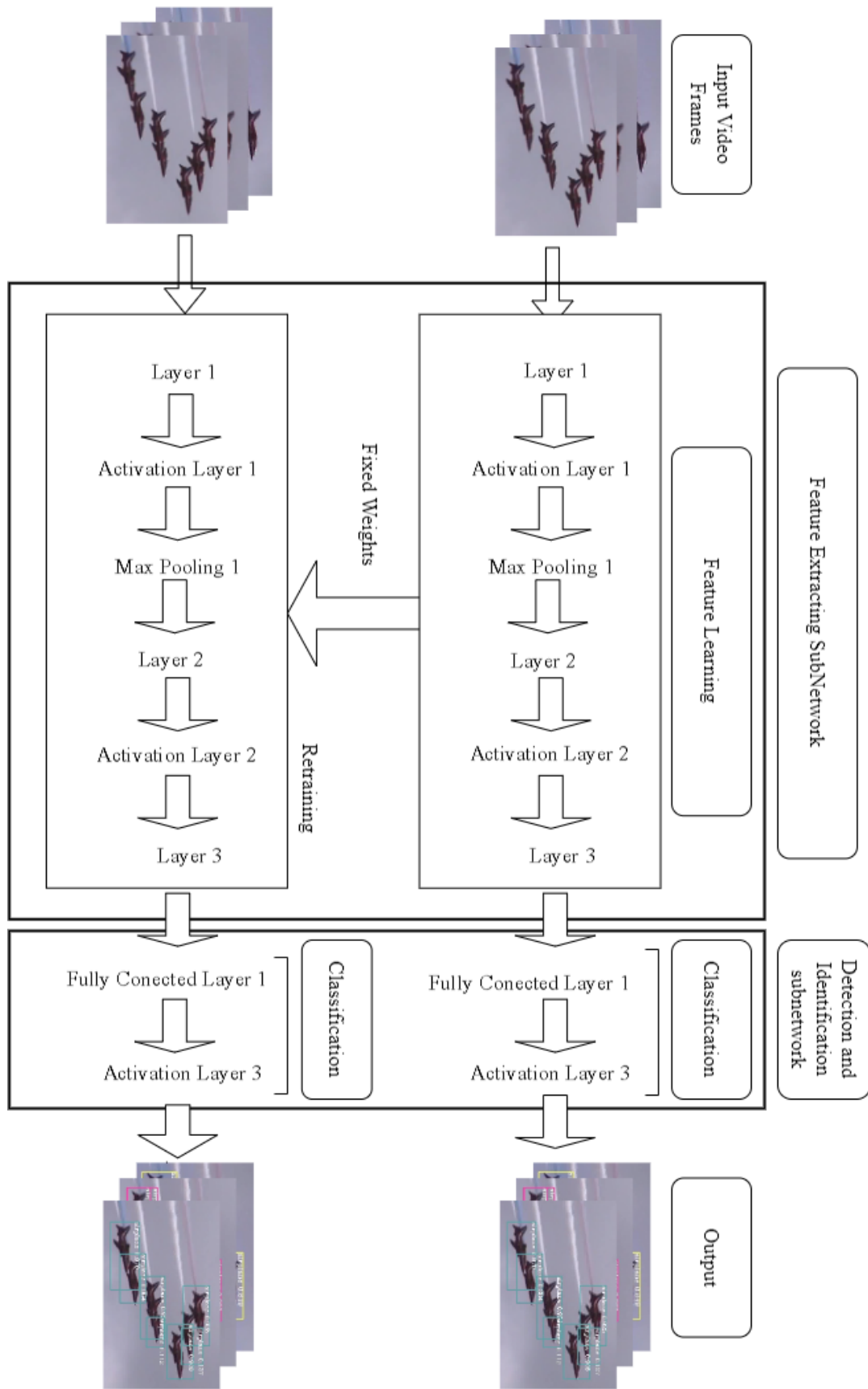


Figure 3.1: Architecture of Proposed Work.

After the activation layer, each block is followed by a max-pooling layer which takes the maximum value out of the stride of the 2x2 matrix to further reduce the dimensions of the feature maps to reduce the requirement of the computation power as the max pooling layer does not make changes into the feature maps of the input frame. There is a fully connected layer after the 5 layers, the feature maps of the all the objects present in the input frame are extracted and the neurons of the last layer are connected with a fully connected layer followed by an activation layer, to identify the object which has been detected based on the feature maps of that particular object. The last activation layer is used to output the final predicted class label for the detected object. The last activation layer has 30 neurons in it, where each of the 30 neurons indicates one class among the 30 class labels respectively.

The weights of each neuron and the weights of the neurons for the forward layer are initialized randomly at the start of the training. As we use the SGD training, our model's errors while the training phase gets backpropagated, and the weights are modified accordingly. A total of 120K iterations are performed for training, where the learning rate is reduced after the first 60K iterations. The weights for the feature aggregation are different from the weights that are used for generating feature maps. The weights for the feature aggregation are also initialised randomly at the start of the training only. The weights of the feature maps are assigned in a way that higher weights are given to the feature maps that are of the nearby neighboring frames and lower weights are assigned to the feature maps of the far frames taking the current frame as the reference frame.

The second model that we use in our proposed model is an instance level based method [4]. The architecture of our second model is the same as the architecture that we have followed for our first model that is, it has 5 convolution layers, where every layer is followed by one activation layer and one max pooling layer. A Fully connected layer is connected with the last layer and there is an activation layer at last after the fully connected layer to predict the output. The activation layer also has 30 neurons which indicate one class label each.

The weights of the second model however are initialized with the weights of the first model which has been trained instead of initializing randomly like the first model. The weights that have been initialized are retrained again and are fined tuned based on the instance level method [4]. The method that we use in our second model for the feature warping is done by calculating the similarity between the proposals that are generated by the RPN using the cosine similarity. The range of the cosine similarity that we get is between -1 to 1 both inclusive. Higher

the value of cosine similarity, the higher the similarity between the proposals generated by the RPN, which means that higher the chances that the proposals are of the same object. Similarly, a lower value of the cosine similarity means that the proposals that are generated are less similar, which indicates that the generated proposals are more likely to be of different objects. The second model being a semantic level model, the proposals that are generated are more likely to be robust against changes in the appearance of the objects.

The reason for using the models of two different methods is that training our model using the pixel level method[2] makes it robust to identify the objects that are non-rigid and are smaller in size. Fine tuning our approach with the semantic level method[4] makes our model more robust towards rigid objects and robust towards change in the appearance of the objects on the occasion of partial or full occlusion as shown in the Figure 3.2

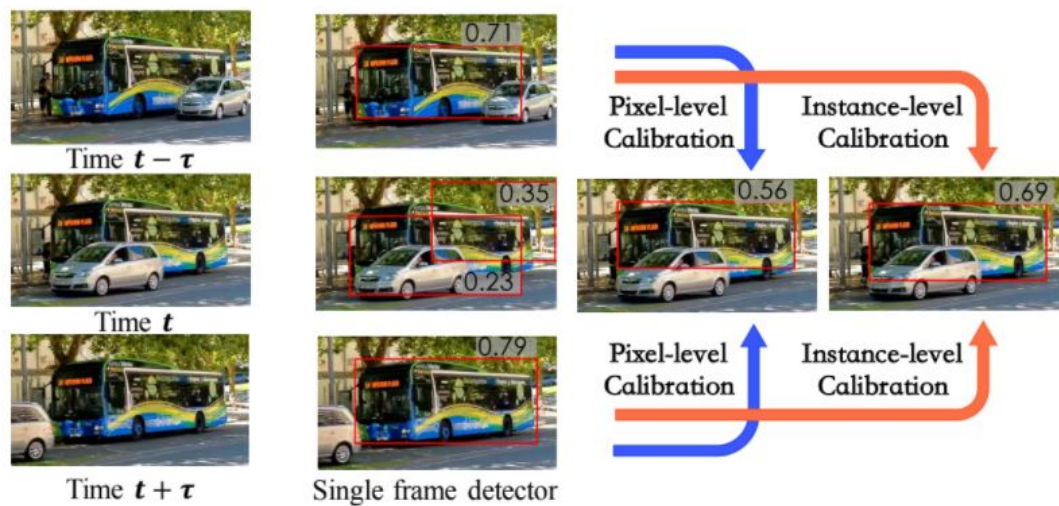


Figure 3.2: Comparison of accuracy for partial occlusion[5].

From the Figure 3.2, one can see that the pixel-level method is only able to detect the bus till the height of the car. Still, the instance level method is able to detect the bus in its entirety with high accuracy, even when the car is standing in front of the bus. This is the reason that we have chosen an instance-level method as our final model for predicting class labels after fine tuning the weights obtained after the training of the pixel-level method.

Feature Warping and Feature Aggregation are two main components of our proposed method's architecture.

1. **Feature Warping:** Its task is to warp feature maps from nearby neighbouring frames with feature maps from the current reference frame, as well as

feature maps from the current reference frame, based on motion between the frames.

2. **Feature Aggregation:** Its work is to aggregate the feature maps of the nearby neighboring frames based on the weights which have been warped by the feature warping module, to enhance the feature maps of the current reference frame.

3.2 Feature Warping

The work of the Feature warping module is to warp the feature maps of the frames, based on the motion between the frames.

3.2.1 Feature Warping for Initialising weights

Motivated by DFF[31], given a a neighbor frame I_j and, a reference frame I_i flow field $M_{i \rightarrow j} = F(I_i, I_j)$ can be estimated by a flow network F (For e.g, FlowNet[31]). Feature maps of the nearby neighboring frames are warped according to the flow along with the current reference frame. The warping function for warping the feature maps of the nearby neighboring frames in the reference frame is defined as follows:

$$f_{j \rightarrow i} = W(f_j, M_{i \rightarrow j}) = W(f_j, F(I_i, I_j)) \quad (3.1)$$

where $W(\cdot)$ is the warping function applied to all places that are in the feature maps for every channel, and $f_{j \rightarrow i}$ refers to the feature maps that are being warped from frame j to frame i .

3.2.2 Feature Warping for trained weights

Let $X^f = \{x_1^f, x_2^f, \dots, x_n^f\}$ denote the generated proposals by using the RPN network of the Faster R-CNN[23]. For any specific pair of generated proposals $\{x_i^f, x_j^f\}$, the similarity between the generated proposals is calculated using the generalized cosine similarity.

$$w_{ij}^{kl} = \phi(\mathbf{x}_i^k)^T \psi(\mathbf{x}_j^l) \quad (3.2)$$

where $\phi(\cdot)$ and $\psi(\cdot)$ are general transformation functions.

3.3 Feature Aggregation

The work of the feature aggregation module is to enhance the feature maps of the objects for the current reference frame by aggregating its feature maps with the feature maps of the nearby neighboring frames.

3.3.1 Feature Aggregation for Initialising weights

The current reference frame after obtaining the warp feature maps of nearby neighbouring frames into current reference frame along with the feature maps of the current reference frame now has very diversified information about the object instances in the current frame. For feature aggregation, different weights for different spatial locations are initialized and all the feature channels for the same location share the same spatial weights. The weight maps for warped features $f_{j \rightarrow i}$ is represented by $w_{j \rightarrow i}$. The aggregated feature maps at the reference frames are obtained as follows:

$$\bar{f}_i = \sum_{j=i-K}^{i+K} w_{j \rightarrow i} f_{j \rightarrow i} \quad (3.3)$$

where K specifies the number of frames that we consider as nearby neighboring frames for feature warping and aggregation into the current reference frame. These aggregated feature maps of the current reference frame are then fed into the detection sub-network to obtain the output bounding boxes over the objects of the current reference frame.

$$y_i = \mathcal{N}_{det}(\bar{f}_i) \quad (3.4)$$

where \mathcal{N}_{det} is the detection specific sub-network to generate the output from the generated feature maps.

3.3.2 Feature Aggregation for trained weights

After the semantic similarity between the proposals is calculated, this semantic similarity now serves as guidance for the proposals of the reference frame on how to aggregate the region proposals of the other nearby neighboring frames. The newly obtained region proposals after aggregating the nearby region proposals have richer information than previously it had and is more robust against change

in appearance like motion blur, rare poses, etc. The similarities are normalized with the softmax function to preserve the magnitude of the features after the aggregation. Assuming that we aggregate F frames of video at random (By using the random selection, we can make the model have global information about the video, which can help our model to detect and identify objects that are experiencing occlusion at a time frame t based on the global information), with each frame having N number of region proposals generated in each frame. The aggregated feature for the referenced proposal is defined as follows:

$$\bar{\mathbf{x}}_i^k = \sum_{l \in \Omega} \sum_{j=1}^N w_{ij}^{kl} \mathbf{x}_j^l \quad (3.5)$$

where Ω is the set of randomly selected frame indexes for the aggregation.

3.4 Loss Function

The loss function is the function that computes the distance between the current output of the algorithm and the expected output. It's a method to evaluate how your algorithm models the data. This section discusses about the loss functions that we have used while training both the models in the training phase.

3.4.1 Loss Function for First Model

The loss function that is used in first model is the standard *Log-loss*.

3.4.2 Loss Function for Second Model

The loss function that is used in second model is the standard *L1 loss*.

CHAPTER 4

Experimental setup

In this chapter, we discuss about the datasets that we have used for our experimentation purpose.

4.1 ImageNet VID 2015 Dataset[6]

ImageNet VID dataset [6]. It is a large-scale benchmark dataset for video object detection. Following the protocols in [9, 27], the training and testing of the video object detection model is done on 3,862 video sequences from the training set and 555 video sequences from the validation set, respectively. All the snippets of the dataset are fully annotated and are at frame rates of 25 to 30 fps in general. There are 30 object categories that we have used for training and testing which are: “airplane, antelope, bear, bicycle, bird, bus, car, cattle, dog, domestic_cat, elephant, fox, giant_panda, hamster, horse, lion, lizard, monkey, motorcycle, rabbit, red_panda, sheep, snake, squirrel, tiger, train, turtle, watercraft, whale, zebra”.

4.2 YouTube-8M Dataset[7]

YouTube-8M Dataset[7] It is a large-scale labeled video dataset that consists of millions of YouTube videos. It has precomputed features for all the videos containing billions of frames. It consists of 6.1 Million IDs and 350,000 hours of videos for training and testing. We have tested our approach on 30 categories which are: “airplane, antelope, bear, bicycle, bird, bus, car, cattle, dog, domestic_cat, elephant, fox, giant_panda, hamster, horse, lion, lizard, monkey, motorcycle, rabbit, red_panda, sheep, snake, squirrel, tiger, train, turtle, watercraft, whale, zebra”.

4.3 Different types of motions in videos

The Ground truth objects in the videos can be categorized into three categories: slow motion, medium motion, and fast motion according to the motion speed for better analysis. An Object's speed can be measured by an indication known as "Motion IoU" which can be measured by averaging the IoU(Intersection-over-union) scores with the same instances but in the nearby neighboring frames. Higher the motion IoU means slower the object moves and similarly lower the motion IoU means the object moves faster.

Object motions are divided as follows into slow, medium, and fast motion. They are categorized as follows:

- **Slow-motion:** motion IoU score > 0.9 .
- **Medium-motion:** motion IoU ≥ 0.7 and motion IoU ≤ 0.9 .
- **Fast-motion:** motion IoU < 0.7 .

Examples of various groups for the different motions are shown in Figure 4.1. For a detailed analysis, the mean average precision(mAP) is calculated for slow, medium, and fast motion along with the mean average precision(mAP) of the entire video.

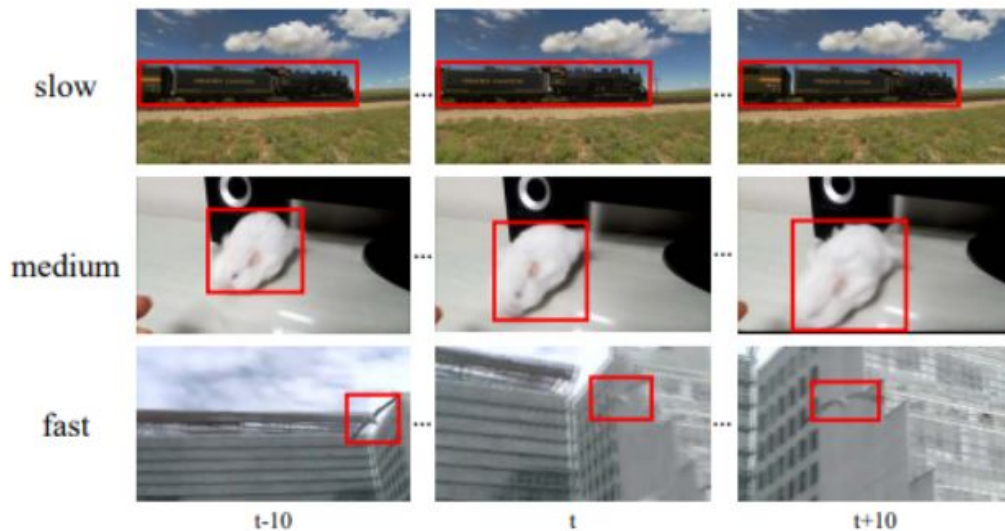


Figure 4.1: Video snippets of object instances having slow, medium, and fast motion[2].

CHAPTER 5

Implementation Details

In this chapter, we discuss in detail the architecture of the feature network that is used to generate and extract feature maps from the input video sequence frame, the detection network that is used to identify the object based on the feature maps generated, and at last talks about how the training is conducted for both the models.

5.1 Implementation Details of First Model

5.1.1 Feature network

ResNet-101[17] and Aligned-Inception-Resnet[21] is taken as the feature network in the first model. The nature of both the models is changed slightly for object detection purposes. The ending average pooling layer and FC layer are removed for object detection purposes. The effective stride of the last block is changed from 32 to 16 to increase the feature resolution as followed in[24, 26]. At the beginning of the last block, the stride is changed from 2 to 1. To reduce the feature dimensions to 1024, a 3 x 3 convolution which is initialized randomly is applied on the top.

5.1.2 Detection network

R-FCN[13] is used for detection following the design as followed in[33]. There are total 1024-d feature maps, on top of which RPN and R-FCN subnetworks are applied where the RPN sub-network connects the first 512-d feature maps and R-FCN connects the last 512-d feature maps respectively. 9 anchors (3 scales and 3 aspect ratios) are used in RPN which generates 300 proposals for each and every image.

5.1.3 Training

Following the approach followed by [30], both the training sets of ImageNet DET and ImageNet VID 2015 Dataset [6] are used. Training is performed in two stages. In the first stage, feature and detection networks are trained on the ImageNet DET training set using the annotations same as in the ImageNet VID 2015 Dataset [6] training set which is 30 categories. One image is used for each mini-batch of SGD training. 120K iterations are performed on 2 GPUS, each GPU holding one mini-batch. For the first 80K iterations, the learning rate is 10^{-3} while for the remaining 40K iterations the learning rate is taken as 10^{-4} . In phase two, the entire FGFA[2] model is trained on the training set, where the weights learned from the first phase are used to initialize the feature and detection networks. 2 GPUs are used in the second phase also. A total of 60K iterations are performed in which, the learning rate is 10^{-3} for the first 40K iterations and 10^{-4} for the last 20K iterations. During the training and inference, all the images are resized to a shorter side of 300 pixels and a shorter side of 600 pixels for the feature network and flow network, respectively.

5.2 Implementation Details of Second Model

5.2.1 Feature Network

ResNet-101[17] is used as the backbone for the feature network in the second architecture. For the final results ResNeXt-101-32 * 4d [34] is used. The effective stride of the last block is changed from 32 to 16 to increase the feature resolution as followed in [24, 26].

5.2.2 Detection Network

RPN is applied to conv4's output. Three scales and three aspect ratios are employed as anchors. The output of conv5 is then subjected to Fast R-CNN to detect and identify the objects that are present in the input video frame based on the proposals generated. On the RoI pooled features, we apply two FC layers, followed by bounding box regression and classification.

5.2.3 Training

Weights obtained after training the first model are taken as pre-trained weights to initialize the backbone networks. SGD training is performed with a total batch size of 4 on 2 GPUs for a total of 220K iterations. $2.5 * 10^{-4}$ is the initial learning rate, which is divided by 10 at the 110K and 165K iterations. For training, one training frame is combined with two random frames from the same video (identical frames for the VID dataset). For inference, K frames from the same video are sampled alongside the inference frame. The photos are scaled to a shorter side of 600 pixels in both training and inference. The runtime of the model is as follows:

$$r = 1 + \frac{(2K + 1) \cdot (\mathcal{O}(\mathcal{F}) + \mathcal{O}(\mathcal{E}) + \mathcal{O}(\mathcal{W}))}{\mathcal{O}(\mathcal{N}_{feat}) + \mathcal{O}(\mathcal{N}_{det})} \quad (5.1)$$

Where F denotes the time for feature map extraction, E denotes the embedding feature maps, W denotes the time for warping the feature maps, N_{feat} denotes the time for updating the feature buffer and N_{det} denotes the time taken for detection.

CHAPTER 6

Experimental Results and Analysis

This chapter discusses the results that we have obtained while performing the testing of our proposed approach on the ImageNet VID 2015 dataset [6] and the YouTube-8M dataset [7]. The results obtained are as follows.

6.1 ImageNet VID 2015 Dataset

Figure 6.1 shows the visualized results of our proposed approach along with three other methods namely FGFA[2], SELSA[4], and MANet[5]. From the Figure 6.1 one can see that results of our proposed approach are better than the other three models not only when only one object is present in the input video frame, but also when multiple objects are present even in the case of partial occlusion as well.

For the single object present in the input video frame, present in the third row of the Figure 6.1, one can see that in the input frame, one buffalo is present which is predicted as a bear by FGFA[2] which is identified correctly by our proposed approach.

Similarly, one can verify for the multiple objects by looking at the second row of the Figure 6.1, where one can see that in the input video frame there is a total of 7 airplanes, while for the results, FGFA[2] and MANet[5] both have identified 8 objects where one region between two airplanes is also misclassified as an airplane. SELSA[4] has identified a total of 7 objects as airplanes but the last airplane is not identified, while our approach identifies all the 7 objects as airplanes with high accuracy.



Figure 6.1: Figure shows the visuals of the results with three other state-of-the-art methods namely FGFA[2], SELSA[4], and MANet[5]. (a) shows the input test frame of the video [6]. (b) shows the output results of the FGFA[2] model. (c) show the output results of SELSA[4] model. (d) shows the output results of MANet[5] model. (e) shows the output results of our approach.

Along with the visualization on a few of the categories as shown in Figure 6.1, we also provide the mAP(%) for all the 30 categories on which we have trained our proposed approach's model in Table 6.1. The table also shows and compares the output of 30 categories of our approach along with the other three methods that are FGFA[2], SELSA[4], and MANet[5] which we have taken as a reference to compare the results of our proposed model's approach.

From the table, we can see that our approach outperforms the other three

Table 6.1: mAP(%) values of all 30 categories for ImageNet VID 2015 Dataset [6].

Categories	FGFA[2]	SELSA[4]	MANet[5]	Our Approach
airplane	89.4	87.92	90.1	91.13
antelope	85.1	84.21	87.3	84.22
bear	83.9	93.51	83.4	91.42
bicycle	69.8	68.90	70.9	73.62
bird	73.5	73.69	73.0	76.83
bus	79.0	77.40	75.6	83.48
car	60.6	59.49	62.0	65.06
cattle	70.7	86.10	74.0	86.07
dog	72.5	84.42	73.3	84.66
d_cat	84.3	87.81	85.3	93.96
elephant	79.9	82.0	79.6	83.95
fox	89.8	94.24	91.6	97.07
g_panda	81.0	79.10	83.5	84.39
hamster	93.3	98.42	96.5	98.73
horse	72.3	81.23	94.5	81.43
lion	50.5	70.57	70.5	80.23
lizard	80.8	83.0	82.0	83.56
monkey	82.3	53.07	54.4	59.79
motorcycle	83.0	87.74	81.6	90.40
rabbit	72.7	83.64	67.0	81.84
r_panda	84.0	89.08	89.3	89.04
sheep	57.8	57.31	73.3	58.40
snake	77.1	66.93	77.4	76.02
squirrel	55.8	54.02	54.3	59.70
tiger	91.95	86.0	91.9	90.26
train	83.8	81.66	82.9	86.21
turtle	83.3	81.54	80.3	82.45
watercraft	68.7	66.04	69.3	69.70
whale	75.9	60.24	75.4	77.23
zebra	91.1	95.54	92.4	97.06
Average mAP(%)	76.3	80.25	78.1	81.93

methods in 19 categories and gives the best results in **19 categories out of 30 categories**. The last row of the table shows the Average mAP(%) of all the 30 categories for all the four methods.

The table 6.2 shows the comparison of the results obtained by our proposed approach in comparison with the other three methods that are FGFA[2] which is a pixel-level based method, SELSA[4] which is an instance-level based method and, MANet[5] which is a combination of pixel-level and instance-level based methods on ImageNet VID 2015 validation dataset [6].

Table 6.2: Accuracy Table for ImageNet VID 2015 Dataset [6].

mAP	FGFA [2]	SELSA [4]	MANet [5]	Our Approach
mAP(%) [1.0]	76.3	80.25	78.1	81.93
mAP(%) slow [0.95]	83.5	86.91	84.9	87.80
mAP(%) medium [0.9]	75.8	77.94	76.8	80.32
mAP(%) fast [0.7]	57.6	61.38	54.37	64.95

For the better analysis of our proposed approach, we compute mAP(%) over the entire video, mAP(%) of slow moving objects, mAP(%) of medium moving objects, and mAP(%) of fast moving objects based on motion IoU score. From the table 6.2, we can see that our approach outperforms SELSA[4] by **2.1%** for the entire video sequence, and our approach outperforms SELSA[4] by **0.9%** in slow-moving objects motion, our method outperforms SELSA[4] by almost **3%** for medium moving objects and outperforms SELSA[4] by almost **5.8%** for fast moving objects as shown in Table 6.2

The confusion matrix in Figure 6.2 shows the accuracy of our method for every 30 categories on which we have trained our model when tested on the ImageNet VID 2015 validation dataset [6]. It also shows the percentage of misclassification that our approach has made considering the 30 categories.

From the Figure 6.2, we can see that our approach shows good results in almost all the 30 categories on which we have trained our model. From the results, we can see that our model performs less number of misclassifications even in the case of change in appearance like partial occlusion.

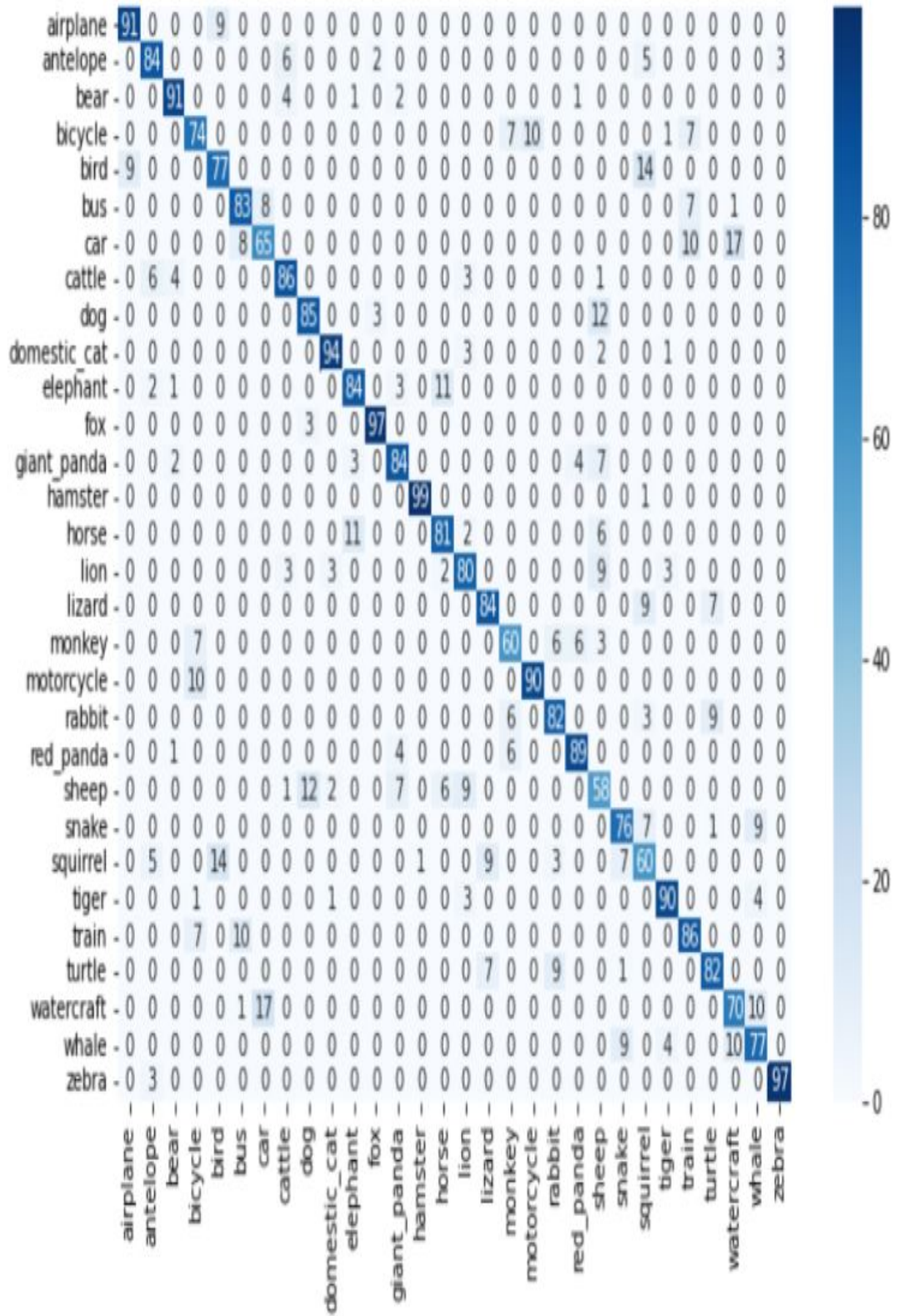


Figure 6.2: Confusion matrix showing the accuracy of all the 30 categories on ImageNet VID 2015 Dataset [6].

Comparison chart Figure 6.3 indicates the motion IoU score on the x-axis and represents the mean average precision(mAP)(%) on the y-axis respectively, where point at 0.7 on x-axis indicates mAP(%) of fast motion, point at 0.9 on x-axis indicates mAP(%) of medium motion, point at 0.95 on x-axis indicates mAP(%) for slow motion and point at 1.0 on x-axis indicates mAP on entire video of ImageNet VID 2015 Dataset [6].

In the Figure 6.3 the red line indicates the results of our approach, the blue line indicates the results of the FGFA[2] method, the orange line shows the results of SELSA[4], and the green line shows the results obtained by the MANet method. From the Figure 6.3 we can see that our approach performs better than all the other three methods, and shows significant improvement in the fast moving objects video sequences.

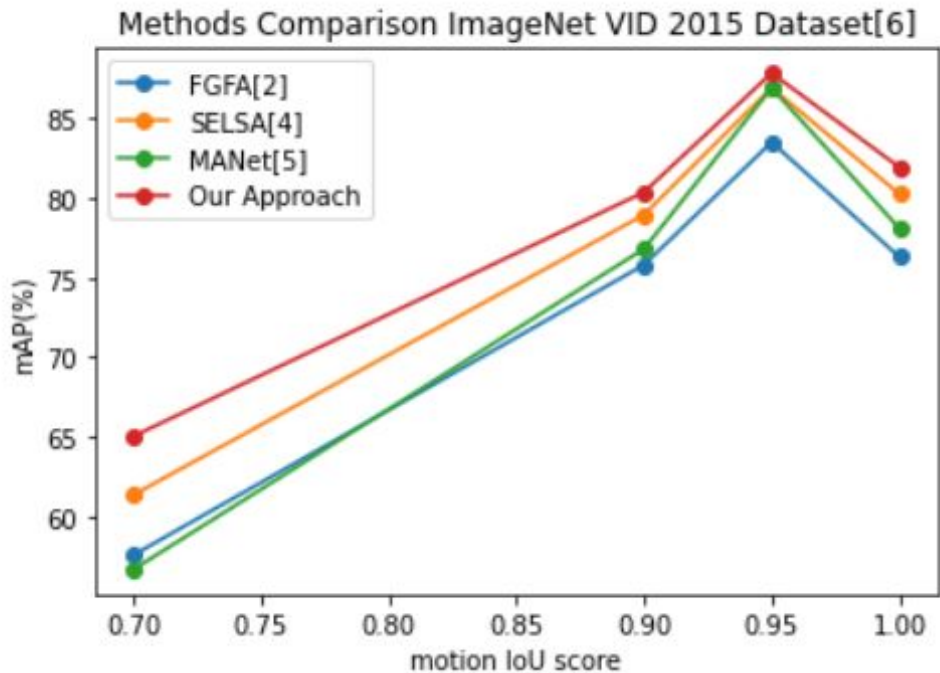


Figure 6.3: Accuracy comparison of proposed method result with other state-of-the-art[2, 4, 5] methods for ImageNet VID 2015 Dataset [6].

6.2 YouTube-8M Dataset

From the Figure 6.4, one can see that our proposed approach shows better results than the results shown from the other three methods - FGFA[2], SELSA[4], and MANet[5]. From the Figure 6.4, one can see that in row one, there are three elephants in the input video frame, where FGFA[2] and MANet[5] identifies only

two elephants, while SELSA[4] and our proposed approach identifies all the three elephants but our proposed approach shows higher accuracy than SELSA[4].

From the Figure 6.4, third row one can see that there are a total of 4 zebras in the input video frame, and FGFA[2] identifies only one object of the input frame, which it misclassified as antelope. Similarly, MANet[5] also identifies only one object of the input video frame, while SELSA[4] identifies three out of the four zebras that are present in the input video frame. While our proposed approach identifies all the four zebras that are present in the input video frame.

The fourth row in the Figure 6.4 is a example of motion blur input sequence that has two birds in the input video frame. FGFA[2] identifies only one bird out of the two objects present, while SELSA[4] and MANet[5] identifies both the birds that are present in the input, but misclassified one object as another object. While our proposed approach detects two objects and identifies them as birds given the input video frame.

Along with the visualization on a few of the categories as shown in Figure 6.4, we also provide the mAP(%) for all the 30 categories on which we have trained our proposed approach's model in Table 6.3.

The table also shows and compares the output of 30 categories of our approach along with the other three methods that are FGFA[2], SELSA[4], and MANet[5] which we have taken as a reference to compare the results of our proposed model's approach. From the table, we can see that our approach outperforms the other three methods in 20 categories and gives the best results in **20 categories out of 30 categories**. The last row of the table shows the Average mAP(%) of all the 30 categories for all the four methods.

The confusion matrix in Figure 6.5 shows the accuracy of our method for every 30 categories on which we have trained our model when tested on the YouTube-8M dataset [7]. It also shows the percentage of misclassification that our approach has made considering the 30 categories.

From the Figure 6.5, we can see that our approach gives good results in almost all the 30 categories on which we have trained our model. From the results, one can see that our model performs less number of misclassifications even in the case of changes in appearance like partial occlusion.

The table 6.4 shows the comparison of the results obtained by our proposed approach in comparison with the other three methods that are FGFA[2] which is an pixel-level based method, SELSA[4] which is a instance-level based method, MANet which is a combination of pixel-level and instance-level based methods on YouTube-8M Dataset [7].

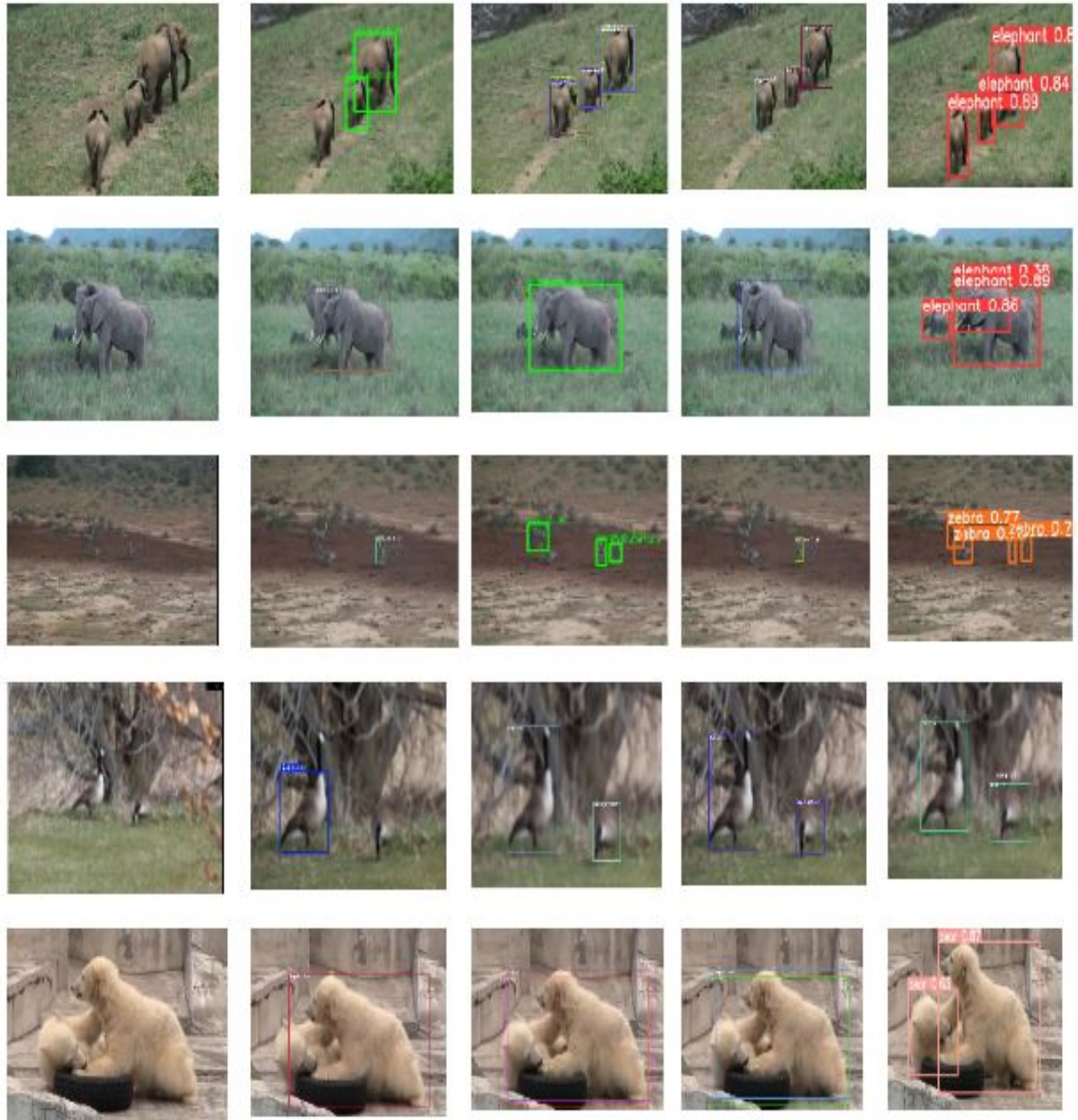


Figure 6.4: Figure shows the visuals of the results with three other state-of-the-art methods namely FGFA[2], SELSA[4], and MANet[5]. (a) shows the input test frame of the video [7]. (b) shows the output results of the FGFA[2] model. (c) show the output results of SELSA[4] model. (d) shows the output results of MANet[5] model. (e) shows the output results of our approach.

For better analysis of our proposed approach, we compute mAP(%) over the entire video, mAP(%) of slow moving objects, mAP(%) of medium moving objects, and mAP(%) of fast moving objects based on motion IoU score. From the table 6.4, we can see that our approach outperforms SELSA[4] by **4.5%** for the entire video sequence, and our approach outperforms SELSA[4] by **1.1%** in slow-moving objects motion, our method outperforms SELSA[4] by almost **1.8%** for medium moving objects and outperforms SELSA[4] by almost **8%** for fast mov-

Table 6.3: mAP(%) values of all 30 categories for YouTube-8M Dataset [7].

Categories	FGFA[2]	SELSA[4]	MANet[5]	Our Approach
airplane	87.92	84.02	92.1	91.10
antelope	80.1	86.21	84.0	84.15
bear	80.9	90.51	83.4	93.45
bicycle	65.8	70.90	69.9	74.62
bird	74.5	73.69	73.0	76.83
bus	79.0	82.40	75.6	83.48
car	59.5	63.2	58.45	65.06
cattle	70.7	86.10	74.0	85.6
dog	84.62	76.9	73.3	84.66
d_cat	84.3	87.81	85.3	93.96
elephant	74.16	84.16	79.6	83.45
fox	90.8	94.24	91.6	98.43
g_panda	81.0	82.7	83.5	84.39
hamster	93.3	98.42	96.5	98.73
horse	72.3	80.23	94.5	79.67
lion	49.5	75.69	70.5	82.96
lizard	83.8	81.75	82.75	83.56
monkey	82.3	53.07	54.4	59.79
motorcycle	84.0	85.74	83.6	92.40
rabbit	83.60	70.4	69.71	80.84
r_panda	82.75	89.08	86.3	90.45
sheep	60.8	57.31	73.3	59.40
snake	72.68	69.93	73.4	76.02
squirrel	52.3	56.82	54.3	58.12
tiger	85.45	89.0	86.75	92.26
train	80.8	86.6	82.7	86.81
turtle	85.3	81.54	82.3	84.95
watercraft	67.7	68.04	65.3	70.70
whale	75.9	60.24	75.4	77.23
zebra	89.1	96.76	91.76	97.06
Average mAP(%)	77.16	78.78	75.48	82.33

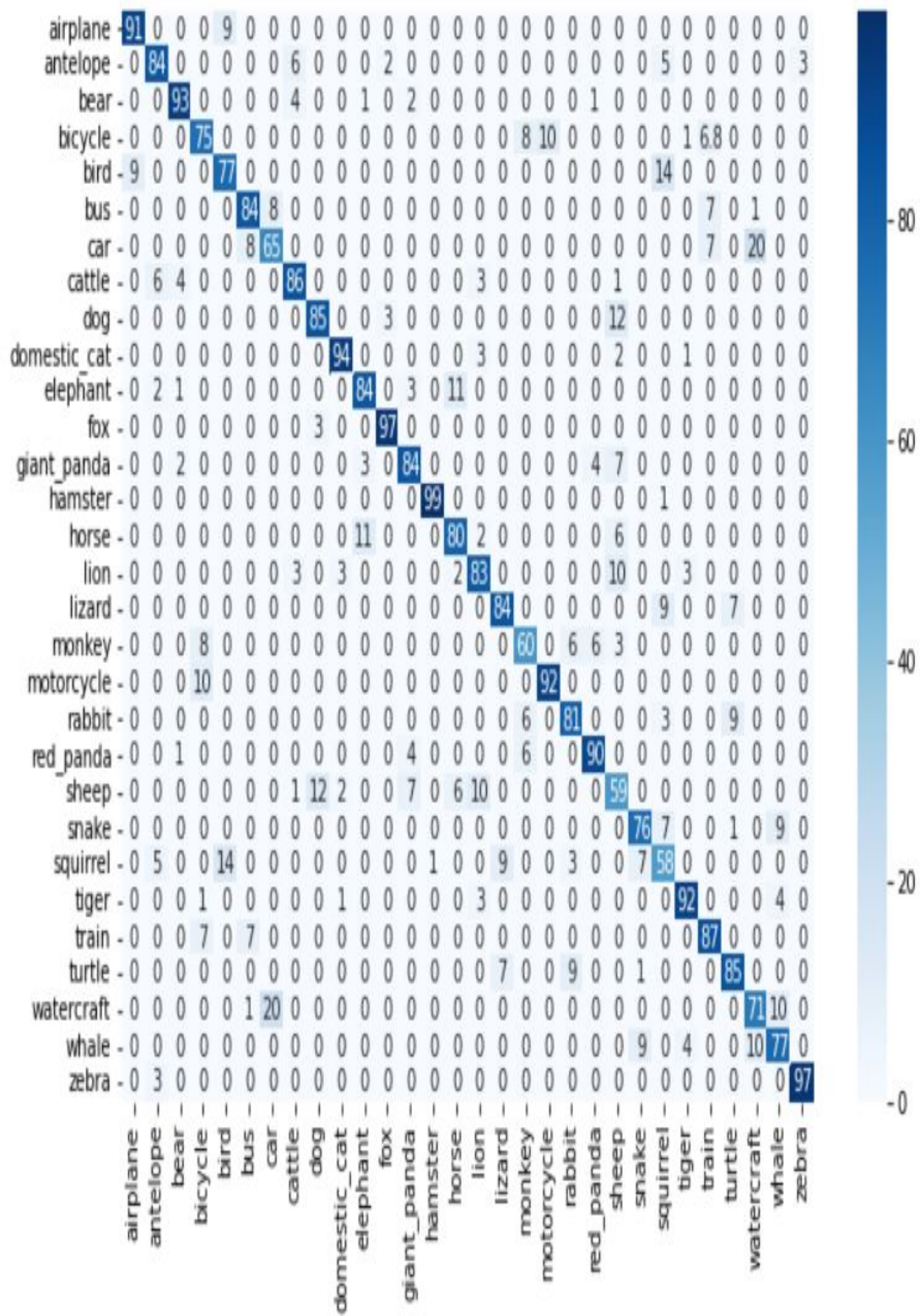


Figure 6.5: Confusion matrix showing the accuracy of all the 30 categories for YouTube-8M Dataset [7].

Table 6.4: Accuracy Table for YouTube-8M Dataset [7].

mAP	FGFA [2]	SELSA [4]	MANet [5]	Our Approach
mAP(%) [1.0]	77.16	78.78	75.48	82.33
mAP(%) slow [0.95]	85.9	86.91	84.9	87.83
mAP(%) medium [0.9]	73.9	78.94	76.8	80.36
mAP(%) fast [0.7]	55.64	60.25	56.7	65.05

ing objects as shown in Table 6.4.

Comparison chart in Figure 6.6 indicates the motion IoU score on the x-axis and represents the mean average precision(mAP)(%) on the y-axis respectively, where point at 0.7 on x-axis indicates mAP(%) of fast motion, point at 0.9 on x-axis indicates mAP(%) of medium motion, point at 0.95 on x-axis indicates mAP(%) for slow motion and point at 1.0 on x-axis indicates mAP on entire video of ImageNet VID 2015 Dataset [6].

In the Figure 6.6 the red line indicates the results of our approach, the blue line indicates the results of the FGFA[2] method, the orange line shows the results of SELSA[4], and the green line shows the results obtained by the MANet[5] method.

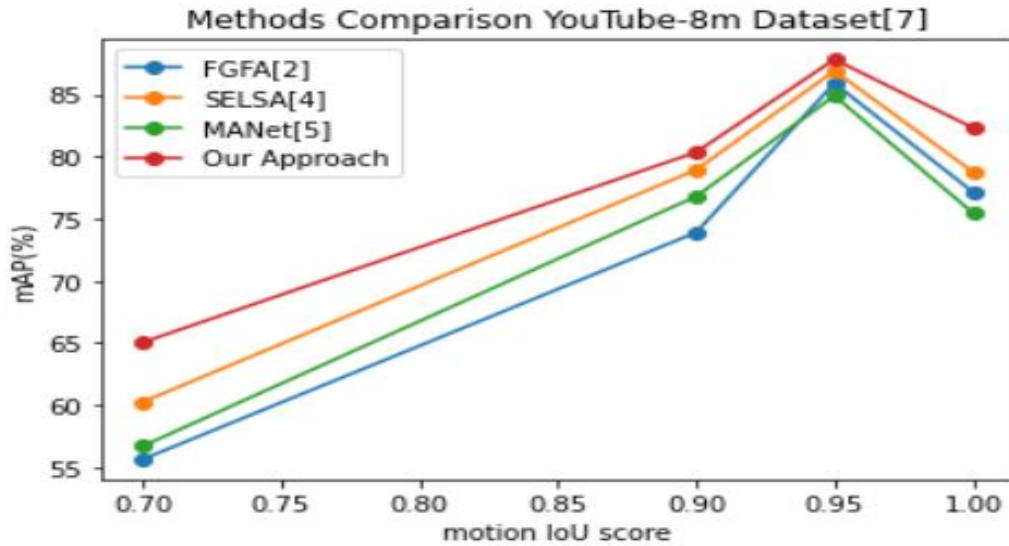


Figure 6.6: Accuracy comparison of proposed method result with other state-of-the-art methods[2, 4, 5] for YouTube-8M Dataset [7].

From the Figure 6.6 we can see that our approach performs better than all the other three methods, and shows significant improvement in the fast moving objects video sequences.

CHAPTER 7

Conclusions and Future Works

7.1 Conclusion

In this work, we have proposed a new method for the Video object detection and identification problem by the taking transfer learning approach on two different methods into consideration for video object detection. Instead of relying fully on the weights of optical flow or full sequence level feature aggregation, we have tried to combine the knowledge of optical flow and sequence level information for feature aggregation and detection. The aggregation performed is done on the proposal level instead of feature maps using the weights trained on both pixel-level method and semantic level method, which makes our method more robust towards motion blur and occlusion. Results and Analysis show that the transfer learning method is effective in case of motion blur and occlusion, outperforms previous methods, and gives the best performance in the instance-level video object detection without any need for post-processing methods/techniques like Seq-NMS.

7.2 Future Works

There is still a large room for improvement in rare poses and fast motion videos. We can make use of relations between the same object in different frames instead of optical flow to get better and enhanced feature maps for better accuracy. We can even detect the edges of the objects before detecting the object, which will make the model robust towards various illumination conditions thus helping to improve the accuracy. For calculating the similarity between the generated proposals for the second model, instead of using cosine similarity, similarities measures that are motion invariant can be used to make the model more robust towards various motion conditions. We believe these open questions will inspire more future work.

References

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.
- [2] X. Zhu, Y. Wang, J. Dai, L. Yuan, and W. Yichen. Flow-guided feature aggregation for video object detection. In *International Conference on Computer Vision*, pages 408–417, 2017.
- [3] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei. Relation distillation networks for video object detection. In *International Conference on Computer Vision*, pages 7023–7032, 2019.
- [4] H. Wu, Y. Chen, N. Wang, and Z. Zang. Sequence level semantics aggregation for video object detection. In *International Conference on Computer Vision*, pages 9217–9225, 2019.
- [5] S. Wang, Y. Zhou, J. Yan, and Z. Deng. Fully motion-aware network for video object detection. In *European Conference on Computer Vision*, pages 557–573, 2018.
- [6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla and M. Bernstein, A. Berg, and F.-F. Li. Imagenet large scale visual recognition challenge. In *International Journal of Computer Vision*, pages 211–252, 2015.
- [7] S. A-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. 2016.
- [8] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. In *International Conference on Learning Representations*, 2016.

- [9] Z. Chen, S. Huang, and D. Tao. Context refinement for object detection. In *European Conference on Computer Vision*, pages 71–86, 2018.
- [10] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Neural Information Processing Systems*, pages 379–387, 2016.
- [11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations*, pages 834–848, 2015.
- [12] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inceptionv4, inception-resnet and the impact of residual connections on learning. In *Thirty-First Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 4278–4284, 2017.
- [13] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *International Conference on Computer Vision*, pages 764–773, 2017.
- [14] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. lownet: Learning optical flow with convolutional networks. In *International Conference on Computer Vision*, pages 2758–2766, 2015.
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.
- [16] Y. Wu, K. Zhang, J. Wang, Y. Wang, Q. Wang, and Q. Li. Cdd-net: A context-driven detection network for multiclass object detection. In *IEEE Geoscience and Remote Sensing Letters*, 2020.
- [17] R. Girshick. Fast r-cnn. In *Conference on Computer Vision and Pattern Recognition*, pages 1440–1448, 2015.
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

- [19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 318–327, 2018.
- [20] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, and W. Ouyang. T-cnn: Tubelets with convolutional neural networks for object detection from videos. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 2896–2907, 2016.
- [21] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 817–825, 2016.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, pages 84–90, 2012.
- [23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems*, pages 91–99, 2015.
- [24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector.
- [25] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 1904–1916, 2014.
- [26] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *International Conference on Computer Vision*, pages 6517–6525, 2017.
- [27] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee. Multi-class multi-object tracking using changing point detection. In *European Conference on Computer Vision*, pages 68–83, 2016.
- [28] J. Redmon and A. Farhadi. Yolov3: An incremental improvement.
- [29] W. Han, P. Khorrami, T. Le Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang. Seq-nms for video object detection.
- [30] J. Uijlings, K. Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. In *International journal of computer vision*, pages 154–171, 2013.

- [31] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 4141–4150, 2017.
- [32] Y. Chen, Y. Cao, H. HU, and L. Wang. Memory enhanced global-local aggregation for video object detection. In *Conference on Computer Vision and Pattern Recognition*, pages 10337–10346, 2020.
- [33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [34] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 2349–2358, 2017.
- [35] C.-Y Fu, W. Liu, A. Ranga, A. Tyagi, and A. C Berg. Dssd: Deconvolutional single shot detector.
- [36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, pages 730–734, 2015.
- [37] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 5987–5995, 2017.
- [38] C.-C Chiu and W.-C. Lo. An object detection algorithm with disparity values. In *4th International Conference on Imaging, Signal Processing and Communications*, pages 20–23, 2020.
- [39] K. Jian and S. Gu. Object and contour detection with an architecture-fusion network. In *IEEE 33rd International Conference on Tools with Artificial Intelligence*, pages 910–914, 2021.
- [40] E. Bayhan, Z. Ozkan, M. Namdar, and A. Basgumus. Deep learning based object detection and recognition of unmanned aerial vehicles. In *3rd International Congress on Human-Computer Interaction, Optimization and Robotic*, pages 467–472, 2021.
- [41] Z. Song, Y. Zhang, Y. Liu, K. Yang, and M. Sun. Msfyolo: Feature fusion-based detection for small objects. In *IEEE Latin America Transactions*, pages 823–830, 2022.

- [42] F. Guo, W. Wang, Z. Shen, J. Shen, L. Shao, and D. Tao. Motion-aware rapid video saliency detection. pages 4887–4898, 2020.
- [43] Y. WU, H. ZHANG, YAWEI LI, YIFAN YANG, and DING YUAN. Video object detection guided by object blur evaluation. In *In IEEE Access*, pages 208554–208565, 2020.
- [44] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang. Object detection in videos by high quality object linking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1272–1278, 2020.
- [45] A. Anjum, T. Abdullah, M. F. Tariq, Y. Baltaci, and N. Antonopoulos. Video stream analysis in clouds: An object detection and classification framework for high performance video analytics. In *IEEE Transactions on Cloud Computing*, pages 1152–1167, 2019.
- [46] B. H. Chen, L. F. Shi, and X. Ke. A robust moving object detection in multi-scenario big data for video surveillance. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 982–995, 2019.
- [47] X. Chen, J. Yu, S. Kong, Z. Wu, and L. Wen. Joint anchor-feature refinement for real-time accurate object detection in images and videos. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 594–607, 2021.
- [48] Y. Huang, Q. Jiang, and Y. Qian. A novel method for video moving object detection using improved independent component analysis. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 2217–2230, 2021.
- [49] L. Han, P. Wang, Z. Yin, F. Wang, and H. Li. Class-aware feature aggregation network for video object detection. In *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
- [50] L. Han and Z. Yin. Global memory and local continuity for video object detection. In *IEEE Transactions on Multimedia*, 2022.
- [51] C. Xiao, Q. Yin, X. Ying, R. Li, S. Wu, Miao Li, L. Liu, W. An, and Z. Chen. Dsfnet: Dynamic and static fusion network for moving object detection in satellite videos. In *IEEE Geoscience and Remote Sensing Letters*, 2022.