CK2

Final Year Thesis Progress

Object Detection from Video

by

Rui Peng

Advised by

Prof. Chi-Keung Tang, Professor

Submitted in partial fulfillment

of the requirements for COMP4981H

in the

Department of Computer Science

Hong Kong University of Science and Technology

2015 - 2016

Date of submission: April 19, 2016

Abstract

Object detection from videos is an emerging area in very large scale visual recognition. While previous state-of-the-art image object detection algorithms can be directly employed in video object detection in a frame-by-frame manner to produce acceptable results, this straightforward solution fails to exploit the rich temporal information inherent video input. In this thesis, I propose a comprehensive video object detection pipeline consisting of a number of flexible modules to produce end-to-end video object detection results. Working in tandem with the Fast-RCNN on VGG16 architecture, our pipeline scores a mAP of 42.1% in the ILSVRC2015 "Object Detection from Video" contest track, placing us the fifth in one of the best known worldwide big-data competition in visual recognition. Along the direction of incorporating temporal information, we present the Fusion-Net, a temporal-aware network architecture for detector. Fusion-Net utilizes the convolution layers (or RNN) to operate on feature maps to provide a comprehensive fusion effect of features along the temporal axis. Preliminary experiments show it is capable of pushing the mAP frontier at least by a margin of 0.5%.

Acknowledgments

First of all, I would like express my sincerest gratitude to my advisor Prof. Chi-Keung Tang, for his continuous support and advice throughout the project. It was his faith and patience in me that gave me a positive attitude towards some of our early unsatisfactory results. I could not imagine a better advisor and mentor for my thesis study.

Secondly, I would like to thank my most important teammate, Hengyuan Hu, who worked along with me through many iterations of design and implementation. His brilliance and attention to details drove the progress forward by great amount.

My deep appreciation also goes to Dr. Yu-Wing Tai, Dr. Cewu Lu, Yongyi Lu and Yuxiang Wu, who provided many valuable suggestions during the ILSVRC2015 VID competition.

Contents

Chapter 1

Introduction

1.1 Overview

Image object detection has been a heated research topic in recent years in the computer vision community. It refers to detecting certain categories of objects by providing tight bounding boxes and category labels as annotations to the image. Academic competitions continue to refine and update theory, methods and techniques. These competitions include PASCAL VOC [4], ImageNet [23] and MSCOCO [17] . Subsequent successful and proven methods are RCNN [9], SPPnet [11], OverFeat [24], Fast-RCNN [8].

With the success in above object detection algorithms on images, one natural question to ask is how these image object detection algorithms can be used in the context of video object detection, since after all, video can be simply viewed as many consecutive images. As the advances in image object detection algorithms are made primarily in recent two years, little work has been done exploring the new video object detection context from utilizing effective image object detection algorithms. To make the problem more precise, we choose to follow the formal definition of "Object Detection from Videos" stated in ImageNet 2015 challenge [22], trying to minimize detection errors and get high mAP (mean average precision) across the entire dataset. Since the problem is relatively new, we wish to present a comprehensive video object detection framework that mainly utilizes state-of-the-art image object detection techniques.

Beyond academia, video object detection has a wide range of applications in industries as

well. It is generally used in video compression, video surveillance, vision-based control, human-computer interfaces, medical imaging, augmented reality, and robotics. Additionally, it provides input to higher level vision tasks, such as 3D reconstruction and 3D representation. Hence, developments in video object detection techniques can ultimately boost performances in numerous fields, leading to higher productivity in daily lives.

1.2 Objectives

Our main focus of the project is to develop a comprehensive video object detection system that utilizes state-of-the-art image object detection techniques. Video, despite being the natural combination of images, turns out to be much more than just a stack of images.

First, the quality of video data is far less ideal than images. There are frequent blurry frames in videos, introduced by motion either from camera movement or object movement during capture. While we do not assume blurry inputs to image object detection algorithms, it is no longer reasonable to assume so in a real world video object detection systems. This requires that the detection system need to be resistant to some level of blurriness.

Second, objects in videos often suffers more from occlusion, due to the relative position change from natural movements. This suggests our new video object detection system must be able to learn the temporal information about objects in consecutive video frames.

To address the above points, we propose a comprehensive video object detection framework to address the challenges separately. The framework is:

- designed to have a complete pipeline with video data as input and object annotations as output.
- equipped with proposal generation routines that are more robust to blurry and low quality video frames.
- capable of labeling highly occluded objects by making use of temporal object motion information extracted from analyzing consecutive video frames.
- augmented by powerful pre-processing and post-processing subroutines to improve the quality of input data and rectify output annotations.

1.3 Literature Survey

It has been well understood that video object detection is usually reduced to image problems. Looking back into the road map of image object detection, there are usually two categories of methods: historical methods (before 2010) and recent methods (after 2012). We view the use of deep convolutional neural networks as a key distinction.

1.3.1 Historical Approaches

Historical approaches primarily falls in two categories: feature-based approaches and motion detection approaches.

Towards better visual feature analysis, a class of feature-based approaches are introduced. For instance, shaped-based methods make heavy assumption on the correspondence between the object category and its uniqueness in shape [5] and color-based methods impose high dependency on the underlying color distribution [7]. Such methods may work well for a small set of object categories or specifically on pedestrians, yet a huge amount of prior knowledge on the objects of interest need to be acquired in advance, making contemporary task of detecting numerous categories a challenging job for visual feature design.

Towards improved temporal motion knowledge, a class of motion detection algorithms have also come into perspective. One major accurate estimate of motion comes from a type of global energy frameworks. It is formulated as a minimization problem towards a global energy objective function which is normally solved with a stochastic or deterministic relaxation algorithm. Along that direction, Markov Random Field gives very good estimate of the motion and is used thereafter for motion detection as a key technique [20]. Such methods gives highly accurate estimates of the motion in video, yet it fails to provide suggestions on what objects are in motion, making it possible to generate labels for the moving object.

As traditional object detection problems are usually formulated targeting specific objects (like pedestrian) or only a small number of object categories, the proposed algorithms employs more object specific knowledge. It is easily imagined that such algorithms is less likely to perform well under our problem context, since it is impractical to hand-code features for feature-based

approaches and relate motion information to object category.

1.3.2 Recent Approaches

Recent approaches, in contrast to previous methods, usually make use of advanced deep neural networks such as CNN, or convolutional neural networks. Originally introduced for document recognition by LeCun [16], it has been later used in many more aspects performing a wide range of visual recognition tasks and have proven its capability in a series of visual recognition contests like ImageNet [13]. Thanks to recent advance in GPU computing and vast availability of training data, CNN architecture is widely adopted whenever a data-driven visual feature representation is needed.

When it comes to analysis on videos, we have seen people make use of CNN to attack many real world problems like video segmentation [6], video description [2] and video classification [19]. Yet experiments on video object detection are still scarce. We suspect the main cause comes from the lack of robust image object detection system at the time of their research.

Fortunately, at the time of our project, there has been quite a number of robust and performing image object detection systems [8, 9, 11, 24]. Among them, the most promising system is Fast-RCNN, which has proven its capability of generating rather accurate bounding boxes for a range of 200 object categories [8]. It has been built upon previous successes of RCNN and SPPnet with the extra advantage of having only single training stage and flexibility of updating all layer parameters, which means it can be rather portable and can be easily fine-tuned contain domain specific knowledge of the target dataset. With such advantages, it can be directly employed as a black box for obtaining image level annotations (bounding boxes and object scores). And thus, we see systems like Fast-RCNN as indispensable building blocks in potential video object detection frameworks.

It can be seen from recent approaches that despite a direct solution to video object detection is not available at the moment, we are equipped with necessary tools for getting image level object annotations, which can potentially simplify our video object detection task in great amount.

And as the video object detection problem has not been formulated well until recently, there has not been much study on the temporal motion features in videos. We have high confidence

in the value of those features as well and seek to incorporate them into video object detection systems to work alongside visual features from image level analysis.

Chapter 2

ILSVRC2015 VID

Since our research objectives greatly overlap with the "Object Detection from Video" track of ImageNet 2015 competition [22], we decide to participate in this prestigious competition for a jump start and devote the first semester to building a video object detection system for it.

As the nature of the competition dictates, this stage of our research is to design a system that can deliver high mean average precision (mAP).

2.1 System Design

In order to achieve higher mAP, we design our system to be a pipeline composed of many separate modules, each responsible for a smaller task. In this way, we can make adaptations to each modules and easily evaluate the performance of the overall system. By doing this, we can iteratively improve the mAP of the system greedily.

To start with, we choose a detection framework to be the center of the pipeline.

2.1.1 Fast-RCNN as Framework

As briefly discussed in Section 1.3.1, we have a handful of frameworks to choose from, primarily RCNN, SppNet, OverFeat, Fast-RCNN, Faster-RCNN [21]. At the time of our decision, unfortunately, the best performing framework Faster-RCNN is not publicly available. We choose

Fast-RCNN being the second best framework that is highly mature and adequately fast for our use case.

2.1.2 Pipeline and Modules

As the design of the Fast-RCNN framework suggests, we also need separate proposal generation routines. To address our concerns regarding low quality video data, we decide to incorporate a pre-processing module to remove unwanted artifacts to a satisfying extent. And in order to leverage temporal information in videos, we plan to augment our detection by running some object tracking algorithms after we get our detection results from the Fast-RCNN detector. Then after that, we can process the detection again using some techniques like Non-Maximum Suppression.

Therefore, we present our pipeline as follows:

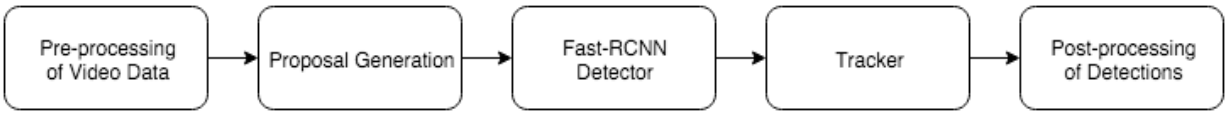

Figure 2-1: System Pipeline

2.1.3 Baseline and Evaluation

Based on a comprehensive system, we construct a baseline of our design so that later innovations can be directly compared to it.

This involves selecting a network architecture for Fast-RCNN detector and a proposal generation algorithm, assuming no attempt on pre-processing, tracking and post-processing.

After a careful exploration (detail discussion see later sections on individual modules), we decide to use VGG16 [27] as the baseline architecture and a widely used selective search [29] as the proposal generation algorithm.

From here, all future performance of altered systems are compared to this baseline and the difference is measured in mAP. Initially the baseline mAP using selective search is 7.9% on

validation set, which is quite low. However, we are determined to score much higher with better proposal generation module.

We adopt a pre-trained model from Fast-RCNN and fine-tune it on video data to be our CNN model. For the evaluation routine, we directly make use of a MATLAB script provided by the ILSVRC2015 devkit [22].

2.2 Exploring individual modules

Our overall pipeline of the system is largely dependent on the performance of each module within the pipeline. Hence by exploiting the room for improvement along each module, we essentially improve the overall effectiveness of the whole system.

2.2.1 Pre-processing

We see great value in pre-processing of video data since we have noticed a non-neglible portion of blurry and low resolution data. Given our prior knowledge that Fast-RCNN detector works quite well on sharp and high resolution data, one straight-forward way to make this video detection challenge easier is to enhance the quality of video data by performing some deblurring and super-resolution techniques.

However, the implementation of such techniques is complex. One good deblurring algoritm [25] has quite impressive sharpening effect yet it imposes significant running time which makes deblurring out of our time budget. Image super-resolution can be done fairly well by some SrCNN networks [3] yet it requires training a separate deep CNN.

Hence, from the above argument, this video enhancing pre-processing direction is deemed unsuitable given the time constraint.

2.2.2 Proposal Generation

Proposal generation is the most important module to the system since it directly feeds candidate bounding boxes to the detector. Suppose we have a very high-performance detector, the recall of the system is almost linearly dependent on the coverage of proposal boxes. With this in mind, much time is spent exploring different proposal generation algorithms along with different configurations and options for the algorithms.

Three algorithms were fully explored: Selective Search, Edge Box [30] and Deep Box [15].

Selective Search mainly utilizes super-pixel technique to greedily find candidate boxes. It empirically works quite well on image object detection systems and has been widely adopted as a standard routine for proposal generation. However, during our experiment with video data, we

discovered that the coverage of the proposals are quite low - around 25% of ground truth boxes considering IOU (Intersection over Union) threshold of 0.5. Our suspicion is that the low quality nature of video data and the frequent small size object worked against this algorithm.

After abandoning Selective Search, we measure the performance of the Edge Box algorithm. The algorithm works extremely well with 96.3% coverage of ground truth under the same IOU setting. Our intuition is that Edge Box is relatively much more robust in blurry images than Selective Search. Hence, we adopted Edge Box to be the proposal generation algorithm in subsequent experiments and use its proposals to re-train the Fast-RCNN network to make the detector acquainted with the objects in our data set.

Deep Box includes a small CNN aiming at re-ranking proposals. Though it's not widely used by many, we decide to examine its effectiveness on video data. After training its CNN on the video data set and applying it to our existing proposals generated by Edge Box, no significant improvements were noticed. The most likely cause might be that Edge Box is already performing quite well and re-ranking simply can not push the limit of proposal qualities.

With the above experiments on proposal generation mostly focusing on coping with the low quality nature of video data, we want to explore further incorporating temporal information in our proposals.

One simple and straight-forward way is to combine proposals from multiple video frames (for example, 2 previous frames + current frame + 2 later frames) and treat them as proposals for the current frame. The key observation is that along consecutive video frames, the objects usually do not move much, hence a potential "object" in the current frame will continue to be a potential "object" in the next frame and vise versa. Simply by this observation and direct implementation, we are able to boost mAP by 3 ~ 5% in many experiments.

2.2.3 Network Architecture

Besides proposal generation, network architecture is another major research direction. Some networks can tailor to video data astonishingly well and some networks can even incorporate temporal information in its neurons.

We have investigated a few network architectures for the system and the summary is as fol-

lows:

VGG16: VGG16 has been adopted widely in the state-of-the-art for image object detection. It is almost perfect for image object detection task. We choose it as our baseline throughout our experiments because it is guaranteed to work well with a number of benchmark statistics available for comparison.

VGG19: VGG19 is a slight upgrade from VGG16 with 3 more convolutional layers and <10% more parameters. It has slight advantage in terms of error rates compared to VGG16. However We choose not to investigate further into VGG19 because it's almost the same as VGG16 but with longer training time to bear.

Early fusion and late fusion [12]: Fusion at CNN architecture level is almost analogous to concatenating CNN features of consecutive frames and perform classification. It seems a bit clumsy and is later beaten by Two-Stream Neural Networks.

Two-Stream Neural Networks [26]: It is an innovative and groundbreaking approach to have a separate temporal channel comprised of optical flow histograms. Two streams of CNN is joined by a fully connected layer to jointly perform classification. This architecture is quite appealing to us since it gives a neat solution to incorporate temporal information at network architecture level. However, we have yet to determine how to port it to detection tasks where we need to label bounding boxes in temporal channel.

HyperColumns [10]: It copes with small objects by aggregating features at multiple levels in CNN. It theoretically can perform better when dealing with small objects and is empirically backed with experiments from Microsoft. However, disk quota seems to be a major limitation since we need to store humongous amount of convolutional feature maps.

In summary, at network architecture level, we have explored quite a number of options. However, we have yet to find one that works better than VGG16 given the time and space constraint.

(In later sections, we have further investigated in more architectures after the competition.)

2.2.4 Tracking

We see tracking as a separate plugin module for rescuing missing detections and rectifying detection scores. After a bit of exploration of common trackers in the recent literature, we managed

to port a quite promising tracker CMT [18] to our system.

In each running of CMT tracker, the tracker takes in current frame and next frame and the object in current frame to produce the location of the object in next frame. It can also report "lose track" if it deems so. The working mechanism of CMT tracker totally based on geometry of key points makes it easy to devise a greedy algorithm that can strictly improve our detection results.

The rough pseudocode for the tracking greedy algorithm can be described as follows:

```
# input: prev_frame, curr_frame, pred = (score, box, cls)
# output: (new_score, new_box, cls)
tracker_frame(prev_frame, curr_frame, pred):
    if score < TRACKER_TH:
        return nil
    next_box = CMT_TRACK(prev_frame.img, next_frame.img, box)
    if next_box is nil: # tracker reports lose track
        return nil
    next_score = score * SCORE_DECAY
    best_overlap_box, best_overlap_score = OVERLAP(curr_frame.boxes,
next_box, IOU_TH)
    if best_overlap_box is nil: # tracker produces new prediction
        return (next_score, next_box, cls)
    else:
        new_score, new_box = TAKE_MAX_SCORE(
(best_overlap_box, best_overlap_score), (next_box, next_score))
        return (new_socre, new_box, cls)
```

Some key thoughts behind the algorithm:

- We use TRACKER_TH to control the entry of predictions, only allowing in very good (with high confident score) predictions. Hence we usually set TRACKER_TH to be 0.8 ˜ 0.9.
- We use a SCORE_DECAY to add timed penalty to the prediction produced by tracker. It is essentially a confidence level we give to the tracker. If we trust our tracker fully, then SCORE_DECAY will be 1, mimicking no penalty. In our experiments, we usually set it to 0.9 ˜ 0.95 since it performs reasonably well.
- We use greedy approach towards the cases where tracker produced prediction is challenging original prediction by detector. We simply treat them equally and take whichever has the higher score if they have a high overlap level (controlled by IOU_TH, usually 0.5).

Hence with this construction of the tracker module, we exhibit a strictly better rectification of predictions, usually with 0.5 ˜ 1% in mAP.

2.2.5 Post-processing of Detections

In the post-processing stage, we mainly utilize a simple technique call Non-Maximum Suppression or NMS. It is used to deal with overlapping predictions and suppress the ones with lower scores.

For example, in the below figure, a prediction of "bird" and a prediction of "airplane" has high overlap level (usually we set a NMS threshold to be 0.3 ˜ 0.4) and we can use the prediction with highest score to suppress others.

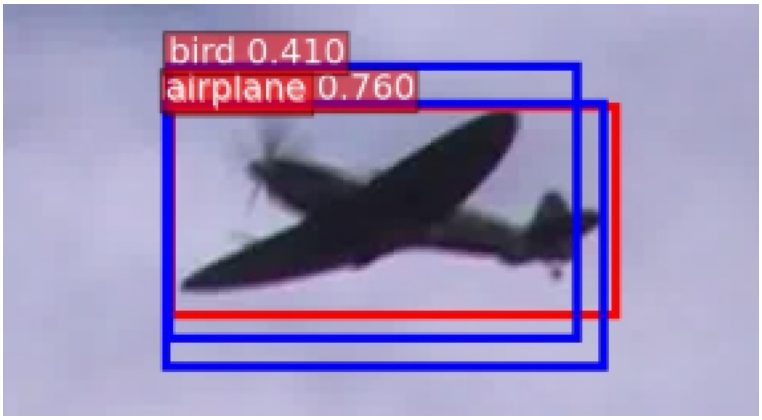

Figure 2-2: Illustration of NMS: low-score "bird" can be suppressed by high-score “airplane”

By such NMS technique across different categories [1], we can further eliminate redundant detections and improve system accuracy and hence boost mAP.

[1] What is worth noticing is that NMS within category is already performed by the detector module, which is a more basic and widely adopted technique.

2.3 Faster-RCNN: a better framework

At the time of our previous research for detection frameworks, Faster-RCNN was an option that was eliminated from the pool since it wasn't publicly available and well integrated. However, around late October when we have finalized our system pipeline based on Fast-RCNN and started investigating on its surrounding modules, the framework was made available, making us hard to port all our system to Faster-RCNN. Despite this fact, we continued to investigate in this framework which claims to be much better and study its benefits.

Basically, Faster-RCNN primarily differs from Fast-RCNN in that it is equipped with a dedicated proposal generation network called RPN (Region Proposal Networks). RPN is structured to perform better on smaller objects and generate more scale-invariant proposals. The detection module is almost the same as Fast-RCNN, in fact shares the convolutional features with RPN, making the network light-weight and easy to train.

In later experiments with Faster-RCNN, when two modules (proposal generation and Fast-RCNN) were replaced, it generates more than 6% boost in mAP, rocketing the number to 46.7%.

It is worth noticing that Faster-RCNN now produces strictly better results on image data and video data, showing the effectiveness of its proposal generation network. With this clear signal, we decided to shift to Faster-RCNN once the competition is over.

2.4 ILSVRC2015 VID results

For the ILSVRC2015 VID competition, we obtained the following results.

In total we have presented 5 submissions with following results[2] :

Submission	Description	Validation mAP (%)	Test mAP (%)
1	Fast-RCNN + EdgeBox × 7	40.1	36.7
2	Fast-RCNN + EdgeBox × 7 + Tracker	40.9	36.7
3	Faster-RCNN + Tracker	46.5	41.5
4	Fast-RCNN + Faster-RCNN (NMS merge)	47.5	41.7
5	Fast-RCNN + Faster-RCNN (NMS merge)	47.9	42.1

Table 2.1: ILSVRC 2015 VID submissions

We are pleased to achieve test mAP of 42.1% and receive a global rank of 5th place. From the above submissions, we notice the relative effectiveness of innovations on different modules. It is seen that framework-wise adoption makes the most significant difference and obviously we need to fully switch to Faster-RCNN. Our tracker module does not perform as we have previously expected and we suspect that it mostly comes from the greedy nature of the algorithm.

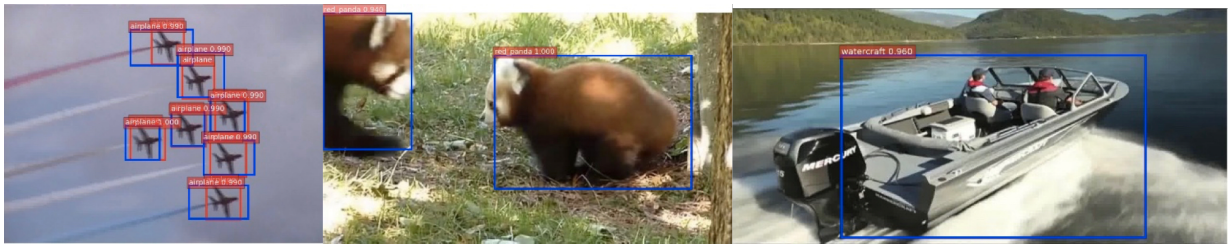

Figure 2-3: Sample predictions from our submission

[2]In the table, “EdgeBox × 7" denotes we use proposals from 7 consecutive frames for the center frame and “NMS merge” denotes we employ cross class Non-Maximum Suppression to combine detections from both systems.

Chapter 3

Incorporating Temporal

After the ILSVRC VID competition, we decide to devote full attention to the unsolved problem – how to incorporate temporal information in a way better than simply aggregating proposals from consecutive frames. Ideally we are looking for ways to devise better network architectural design.

3.1 Faster-RCNN as New Baseline

To compare our new innovations, we need to specify a baseline. We adopt Faster-RCNN based on our previous knowledge that it performs well enough without other supporting modules and it is relatively easy to train.

We have compared two ways for training the baseline network, alternating optimization and end-to-end optimization. The former one involves fixing one half of the network parameters and allow parameters to update for the other half between RPN and detector parts, whereas the latter one employs end-to-end training by specifying a weight between loss from RPN and detector to perform joint optimization. Empirically, the second works slightly better with 1% improvement in mAP. Hence, we take the network trained using second method as our baseline.

Another technique we have employed is sampling on training set. We have sampled every other 30 image frames so as to reduce the amount of data allowing faster convergence and reducing overfitting by removing redundancy in video data. We also run validation on sampled data

set with the same sampling rate.

With the above setup, our baseline model can already achieve a high mAP score of 44.46% on validation set. It is worth noticing that this baseline performance is already good enough for placing before 10th in the VID competition. The motivation of using this as baseline is that if we can devise elegant ways to incorporate temporal information that beats this baseline, we can directly contribute to some novelty in the field of video object detection.

3.2 Optical Flow as Temporal

Given the huge success of Two-Stream [26] architectures in video action recognition, we see optical flow a potentially promising way to encode temporal information. With the inspiration from Two-Stream architecture, we decide to design a similar network on top of Faster-RCNN. We designed and implemented a separate stream with same set of convolutional architecture up to "conv5" in VGG16, similar to what is done in Two-Stream. However, after trying out the architecture, it did not have the high mAP score we anticipated, roughly 2-3% lower than baseline.

To further investigate in the result we had, we question whether optical flow is the right way to encode temporal information in the context of object detection. With the purpose of proving the value of optical flow as temporal representation, we devise two testing network all based on simple adaptations on Faster-RCNN.

3.2.1 One-Stream

We change the dimension of the depth of data in Faster-RCNN from 3 (RGB) to 3 (RGB) + 2 * n (n frames of optical flows along x and y direction). This allows us to stack spatial and temporal information together and feed into the same network. Since this network packs two streams of spatial and temporal information into one stream, we call it One-Stream. We trained the network using the same training configuration as our baseline and got 1-2% lower mAP than baseline.

It was counter-intuitive at first to us that with more information, the network actually performed worse. Yet after careful consideration, we discovered that the network most likely fail to generalize well with two streams of data fed simultaneously into single stream given the different nature of the streams.

3.2.2 Temporal-Stream

In the above experiment, it is possible that optical flow itself is not a good temporal information representation. Hence we go on to investigate if the network can perform well only relying on optical flows. It might sound too challenging only using optical flows to perform detection yet

success stories from Two-Stream do present that a good optical flow stream itself is capable of producing similar performance with the spatial stream in the action recognition task. We feed temporal information (2 * n channels of optical flow) to the Faster-RCNN network and call this experiment Temporal-Stream. It is indeed to our expectation that the Temporal-Stream performs quite poorly, attaining only <10% mAP.

3.2.3 Closer Analysis

We have ruled out the cause from quality of optical flow generation routine as it is obtained from standard OpenCV implementation and used in many successful methods including Two-Stream.

Now with the above experiment in Temporal-Stream network, we have concluded that the use of optical flow as temporal information actually dragged spatial stream down. To further support this, we fedd all zero to One-Stream as optical flow data to simulate no temporal information, the network produced very similar result to the one with real optical flow with less than 0.1% margin. It directly shows that the optical flow temporal information is mostly negligible noise to the system.

All the above experiments essentially prove temporal information as separate stream or channel is not useful. We carefully compare with the situation in Two-Stream paper and get following summary:

- Two-Stream paper targets action recognition task. It is essentially movement classification where the movements encoded in optical flow can be directly related to the category. Imagine the category of "Running", the optical flow shows the movement of arms and legs which is the characteristic of such category according to human perception. Video object detection is different in that the movement of an object only infers the existence of some object yet failing to convey characteristic information about the object. In this sense, optical flow is too broad for the network to infer category labels and thus create difficulty in detection.
- The datasets in two tasks is drastically different. In ILSVRC VID dataset, videos usually contain very little movement hence the optical flow is relatively rare and of small magnitude (see Figure 3-1). However in HMDB51 [14] and UCF101 [28], the datasets used

in Two-Stream paper, movements are generally of much bigger range and scale and the subject occupies most area of the video frames (see Figure 3-2). Such differences make optical flow in VID a lot less ideal for temporal information representation.

Figure 3-1: Sample optical flow (shown in blue arrows) from VID dataset: small movement magnitude (turtle not even moving) + small subject

Figure 3-2: Sample optical flow (shown in blue arrows) from HMDB51 dataset: large movement magnitude + big centered subject

3.3 Temporal in Feature Space

We add temporal information in the hope that the network can join links between temporal information and spatial information. This is straight-forward to humans. We are able to identify object despite partial occlusion and corruption because we can infer object movement from previous and next visual frames. We want the network to have similar abilities – essentially using visual features from nearby frames to make inference on current frame. Based on this assumption, it seems adding temporal connection in feature space more straightforward to the network to make use of.

Experiments from Inside-Outside Net [1] show that one can use Recurrent Neural Networks (RNN) to perform on feature space to add spatial correspondence between pixels. This gives us the inspiration that we can deploy similar techniques like RNN to operate on each feature maps (the filters obtained after "conv5") so that each new feature map on current frame contains the same feature from previous frames. In a sense, this feature map encodes temporal information only relevant to the feature itself, which is quite desirable. However, on the practical standpoint, training 512 RNNs is rather complicated. To simplify training, we chose to use 1×1 convolution layers instead of RNNs to compress feature maps along time axis. Though there are loss in generalization ability switching to simple convolution, we think a working convolutional partial generalization is a preliminary model to test the RNN idea.

3.3.1 Fusion-Net

As for implementation, we design the network that takes in 2 images, second one being the current frame and first one being a past frame with bounded random time difference. The network calculates feature maps for each image and use 2 suites of convolution (1×1 convolution layer + ReLU) to fusion the feature maps into one and use the new one for detection for the second image. We call the network Fusion-Net.

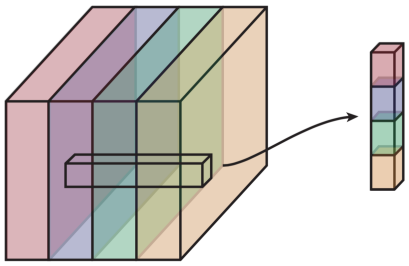

Figure 3-3: Illustration of stacking feature maps along temporal-axis: Each color represent a feature map of the same feature computed using different video frames. The concatenated volume on the right is later fed into 1×1 convolution layers to perform compression.

We employed two training stages for the network. First, we fixed the parameters up to "conv5" to preserve feature extraction and trained for 70K iterations. Then with lower learning rate, we allowed the error to propagate to early convolution layers for another 100K iterations.

Fusion-Net produces pretty decent result on validation set, with a mAP score of 45.02%, which is about 0.5% higher than baseline. Though the difference is still by a small margin, we believe the toy model has quite significant implication on the general "operation on feature map" idea.

Chapter 4

Conclusion

In conclusion, we have achieved our objectives to a major extent.

- We have designed and implemented a comprehensive video object detection pipeline. It is capable of producing end-to-end detections.
- The proposed pipeline either uses Edge Box (in Fast-RCNN) or RPN (in Faster-RCNN) for proposal generation. Either proposal generation routine is highly robust to blurry and low quality video frames.
- The proposed pipeline is able to utilize proposals from consecutive frames to incorporate temporal information in proposal level.
- With Fusion-Net, we have proved the effectiveness of fusion at feature level using 1×1 convolution layers.
- We have developed a novel way to incorporate tracker into post-processing stage that can boost mAP in many situations.

Chapter 5

Future Work

Inspired by many published work by competitors in the ILSVRC and MSCOCO competitions, we find two directions promising to further look into.

- We have proved the effectiveness of 1×1 convolution layers in combining feature maps in time axis by preliminary results. This is only a simplified model from RNN inspired by [1]. It is worth exploring to see whether RNN can generalize the combining function better. We have strong faith in the future in RNN in this case since RNN is the state-of-the-art way of generalizing time series information.
- CNN based tracking by propagating scores is another promising way to go demonstrated by the success in VID competition achieved by CUHK team. Intuitively, a CNN based tracker can recognize high level visual features of detected objects and thus can track the object better than geometry based trackers. In addition, the detection results can provide category information to the tracker which allows the tracker to focus on only one category.

Appendix A

mAP Table

The following table shows the most crucial numbers used for comparing different network architectures.

Network	Training Iterations	validation mAP (%)
Faster-RCNN	70K	44.46
One-Stream	70K	42.34
Temporal-Stream	70K	9.86
Fusion-Net	70K	43.71
Fusion-Net	170K	45.02

Table A.1: Different network architecture mAP scores

Appendix B

Minutes

Meetings with Prof. Tang and teammates in ILSVRC2015 VID team usually take place weekly. Exact meeting records are kept in emails and my personal FYT logs.

Appendix C

Required Hardware and Software

C.1 Hardware

- Computing Servers - CKCPU1, CKCPU2 provided by Prof. Tang
- GPU - 2 Tesla K40C + 2 Titan X
- Hard Disk - >5TB
- Memory - 64GB for CPU, 12GB for GPU on each device

C.2 Software

- Operating System - Ubuntu
- Fast-RCNN public implementation (caffe based)
- Faster-RCNN public implementation (caffe based)
- Proposal Generation Algorithms (Selective Search, Edge Box, Deep Box) public implementation
- OpenCV 2.9
- Python 2.7 (with essential modules like numpy, scipy, matplotlib, etc)
- MATLAB

Bibliography

[1] S. Bell, C. L. Zitnick, K. Bala, and R. B. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," *CoRR*, vol. abs/1512.04143, 2015. [Online]. Available: http://arxiv.org/abs/1512.04143

[2] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *arXiv preprint arXiv:1411.4389*, 2014.

[3] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 184–199.

[4] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[5] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *Journal of intelligent information systems*, vol. 3, no. 3-4, pp. 231–262, 1994.

[6] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen, "Jumpcut: Non-successive mask transfer and interpolation for video cutout," *ACM Trans. Graph. (SIGGRAPH Asia 2015)*, vol. to appear, no. to appear, p. to appear, 2015.

[7] P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 21–27.

[8] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083*, 2015.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 580–587.

[10] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 447–456.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 346–361.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] H. Kuehne, H. Jhuang, R. Stiefelhagen, and T. Serre, "Hmdb51: A large video database for human motion recognition," in *High Performance Computing in Science and Engineering âĂŸ12*. Springer, 2013, pp. 571–582.

[15] W. Kuo, B. Hariharan, and J. Malik, "Deepbox: Learning objectness with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2479–2487.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[18] G. Nebehay and R. Pflugfelder, "Clustering of Static-Adaptive correspondences for deformable object tracking," in *Computer Vision and Pattern Recognition*. IEEE, Jun. 2015.

[19] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," *arXiv preprint arXiv:1503.08909*, 2015.

[20] N. Paragios and G. Tziritas, "Adaptive detection and localization of moving objects in image sequences," *Signal Processing: Image Communication*, vol. 14, no. 4, pp. 277–296, 1999.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, pp. 1–42, April 2015.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, pp. 1–42, 2014.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[25] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 73.

[26] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[27] ——, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[28] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[29] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.

[30] L. Zitnick and P. Dollar, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.