CONTEXT-AWARE COLLABORATIVE OBJECT RECOGNITION
FOR MULTI CAMERA TIME SERIES DATA

A Thesis in
Computer Science and Engineering

by

Philip Wootaek Shin

©2019 Philip Wootaek Shin

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2019
The thesis of Philip Wootaek Shin was reviewed and approved* by the following:

Vijaykrishnan Narayanan  
Distinguished Professor of Computer Science and Engineering  
Thesis Co-Adviser

John M. Sampson  
Assistant Professor of Computer Science and Engineering  
Thesis Co-Adviser

Chita R. Das  
Distinguished Professor of Computer Science and Engineering  
Department Head of Computer Science and Engineering

*Signatures are on file in the Graduate School.
Abstract

Recent research shows that the multi-view system for object recognition outperforms the single-view point system. When viewpoints are added, additional communication cost and cost to deploy the viewpoints are also added. However, prior work has shown that not all of the views are useful, and poor viewpoints can be excluded.

This thesis explores the dynamic context application for a Context-Aware Neural Network. The Context-Aware Neural Network uses Shannon entropy value to acquire likelihood, and this likelihood value to reduce viewpoints in a distributed system. However, reducing viewpoints were done on static image recognition, so the spatial relation between the views and subject is fixed. Expansion to dynamic context is essential since most of the real world is a series of images, rather than a snapshot of the scene. Apart from testing on images of 3D CAD data, this thesis illustrates the generation of 3D CAD data videos, and examines the video analysis of the generated videos using the Context-Aware Neural Network. In this particular setup, relevant objects move with respect to a fixed set of cameras. It is reported that the viewpoints can be reduced by 25-66.7 percent per frame.
# Table of Contents

List of Figures ........................................ v

List of Tables ......................................... vii

Acknowledgements ..................................... viii

1 Introduction ......................................... 1

2 Background .......................................... 3

3 MVCNN-Wavg for testing dynamic context .............. 5
   3.1 Architecture of MVCNN-Wavg ......................... 5
   3.2 Video generation and training ......................... 6

4 Results .............................................. 9
   4.1 Baseline test ......................................... 9
   4.2 Viewpoint reduction on videos ....................... 11
   4.3 Viewpoint reduction on videos with contextual mismatch .... 16

5 Conclusions ......................................... 21

Bibliography ......................................... 23
List of Figures

3.1 The architecture of Context-Aware Convolution Neural Network[1] ......... 5

3.2 Orientation of objects moving in video: (a) shows the 3D orientation of the object and cameras placed, (b) shows orientation facing the front of the object and (c) shows orientation on top of the object. The bottom of the circle is denoted as view0 and from there, counter-clockwise, view1 through view11 ......... 7

3.3 Object movement in every frame in the left diagram and images on the right, showing every 10 frames and four different views of the same object moving. This is to illustrate how the object is moving ......................... 8

3.4 Sample of the videos. (a), (c), and (e) represent the first frame of the video and (b), (d), and (f) represent the sixtieth frame of different 3D CAD objects .......... 8

4.1 Accuracy of Wavgpooling without threshold and with threshold for every frame .. 11

4.2 Every five frames accuracy for Wavgpooling, maxpooling, avgpooling, and Wavgpooling with threshold ......................................................... 12

4.3 Distance from the center and accuracy of different pooling. This is basically the same graph as Figure 4.2, with a different x-axis .......................... 13

4.4 Views are reduced when the object is moving away from the center. A total of 70 videos were evaluated and if less than half (35 or less) have non-zero value on those viewpoints, that specific viewpoint is reduced ............................ 14
4.5 The impact of viewpoint reduction on Wavgpooling. In frame 0, and in every fourth and ninth frame, the likelihood is calculated, so the accuracy of Wavg with threshold and Wavg with rules applied is the same .................................................. 15

4.6 Accuracy of Wavgpooling without threshold and with threshold for every frame . . 17

4.7 Every 5 frames accuracy for Wavgpooling, maxpooling, avgpooling and Wavgpooling with threshold ................................................................. 18

4.8 Distance from the center and accuracy of different pooling. This is basically the same graph as Figure 4.7, with a different x-axis .................................................. 19

4.9 Views reduce when the object is moving away from the center. A total of 70 videos are evaluated. If less than half use the viewpoints, reduction of that specific viewpoint is done .................................................. 20
List of Tables

4.1 Accuracy of single view and multiple view ........................................ 9
4.2 Accuracy tested on generated video dataset ........................................ 10
4.3 Accuracy based on number of context group for maxpooling, avgpooling, and Wavgpooling ......................................................... 10
4.4 Each number inside the table indicates the number of objects that has non-zero entropy value after applying threshold ........................................ 14
4.5 Accuracy tested on generated video dataset without normalization and different context ................................................................. 16
Acknowledgements

I would like to thank Professor Vijaykrishnan Narayanan and Professor Jack Sampson. They provided me guidance in the research and valuable advice during experiments while I wrote my Masters thesis. I would also like to also thank Professor Abhronil Sengupta and Jinhang Choi for giving me feedback in experiments. Last but most importantly, I would also like to thank my parents, Dongkyoo Shin and Inhye Ham. They have been always supportive and will be my best supporters.

This work was supported in part by Semiconductor Research Corporation(SRC) Center for Brain-inspired Computing Enabling Autonomous Intelligence (C-BRIC). The views and findings expressed herein are solely those of the author and do not reflect the opinion or position of the sponsor.
Chapter 1

Introduction

Image recognition has been a field of study for centuries. Deep Neural Networks (DNNs) have flourished after the advent of Alexnet [2]. The rich deep features generated by DNNs have provided better results in image recognition than any other previous methods.

While single-view was previously used in image recognition, many real-world scenarios offer multiple viewpoints. Many viewpoints are used in stereo vision systems, sporting events, and security cameras in stores. To match the needs, research has been conducted regarding Multi-View Convolutional Neural Networks (MVCNN). Rather than a single view of the image, if given multiple views of the same object, DNNs can obtain richer deep features. An MVCNN network was first utilized by building descriptors that are a combination of information from multiple views [3]. Subsequent research has adopted the MVCNN and expanded it to improve the base model [4, 5]. However, the descendant work has not focused on reduction of viewpoints, instead aggregating all viewpoints without considering viewpoint-wise context. There is more information in multi-viewpoint systems, but it is redundant among viewpoints.

Our prior work [1], an MVCNN using weighted average pooling (MVCNN-Wavg), started by considering viewpoint-wise context and reduction of trivial viewpoints. Since there are hardware resources that can perform processing computation of DNNs [6], the endpoint can make decisions with high confidence [7]. By using context grouping and weighted average pooling, MVCNN-Wavg achieved removal of redundant information across viewpoints and finding contex-
tual salience of known viewpoints. However, this work has considered only static image recognition, where the spatial relation between the views and subject is fixed. It has used static, uncorrelated still frames of images.

In a real-world scenario, most stereo systems, sporting events, and security cameras deal with image streams rather than a single image, where relative value is a dynamic property with respect to static placed cameras. As a dynamism, the considered set of objects are in the room, moving in and out of it; which camera has the best view of objects in each frame of the video is a time-varying answer.

This thesis focuses on reducing viewpoints for the stream of images using MVCNN-Wavg and comparing it with MVCNN methods. Moreover, MVCNN-Wavg will be used on moving objects with respect to a fixed set of cameras to examine how many viewpoints can be reduced, considering the spatial relationship and reduction of viewpoints.
Chapter 2

Background

Among different image processing techniques, Convolutional Neural Networks (CNN) have been investigated and used in machine learning and the vision community. CNNs are created to process data that comes in the form of multiple arrays. The key ideas behind CNN are local connections, shared weights, pooling, and usage of multiple layers [8]. Although there were multiple applications of CNNs until the 2000s [9, 10, 11], CNNs were not part of the computer vision and machine learning community mainstream until the advent of Alexnet. This deep convolutional neural network (DNN) was applied to millions of images containing 1000 different classes and halved the error rate of the competing approach at that time [2]. In addition, the parallel computation power from GPUs and its advancement made DNNs a success. DNNs are used as they are most effective in most modern detection and recognition tasks [12, 13, 14, 15].

Distributed deep neural networks (DDNN) use end devices (sensors) to collect data while locally performing inference. Recent work on DDNNs has shown efficient mapping of larger networks onto end nodes. BranchyNet [16] introduced a framework to reduce a large amount of computation at the early termination point with relatively the same accuracy. After early termination, it is possible to lose more confidential nodes. Teerapittayanon et al. [17] offloaded to edge and cloud by using early exit points in the DDNN, followed by aggregates across end devices to perform classification, which would decrease the communication cost between edge and cloud. However, limitations arose during the aggregation process, since it lost context information.
Efforts have been made to use multiple views to offer better recognition performance than a single view. A Multi-View Convolutional Neural Network (MVCNN) was first established by building a descriptor that combined multiple viewpoints to gain better accuracy than a single viewpoint [3]. This was the first approach to use multi-viewpoints using the modelnet dataset [18] to gain better accuracy. Descendant research has been conducted on improving accuracy. Rotation-net [19] jointly predicts object category and viewpoints from each single-view image and associates the object class predictions retrieved from a partial set of multi-view images. Group-view convolutional neural networks (GVCNN) [20] extracts a viewpoint descriptor, groups the views according to discriminative level, and utilizes this grouping to shape representation. The descendant works have increased the overall accuracy more than Su et al. [3]. However, the limitation is that they treat all views equally, resulting in neglect of possible context in each viewpoint.

An MVCNN using weighted average pooling (MVCNN-Wavg) [1] is an architecture that uses context grouping and Shannons entropy [21] to calculate likelihood. The weights are determined by this likelihood calculation and context grouping. Weighted average pooling with threshold is used to reduce the number of less important end devices. It is significant in the removal of redundant information across viewpoints and determining contextual salience of known viewpoints. However, this architecture is focused on static images of viewpoints, so the spatial relation between the views and subject is fixed. In addition, the subject was placed in the center of the viewpoints, which is not applicable in the real world. In reality, the object can be placed at different positions and, instead of a static image, there are dynamic movements in objects with streams of images. This thesis adopts this idea to study the possibility of reducing viewpoints in the stream of videos and applying the dynamism scenario on MVCNN-Wavg.
Chapter 3

MVCNN-Wavg for testing dynamic context

3.1 Architecture of MVCNN-Wavg

MVCNN-Wavg [1] was adapted as a baseline to test the dynamic context. For CNN design on MVCNN-Wavg, Alexnet [2] was deployed. As shown in Figure 3.1, the network takes multiple viewpoints of an image. In implementation, 12 views of different objects are fed into the network. Twelve viewpoints are selected in order to be consistent with the framework of the previous work. The feature extractors have their own viewpoint-wise context. There are similarities between neighboring nodes within the context group, but each node has its own local feature extractor. The weights and biases used are taken from Alexnets weights and biases.
The likelihood estimation is calculated using each views entropy value. The threshold is applied after calculation of the entropy value. If the entropy value is lower than the threshold, we do not send it to the cloud of that particular viewpoint. In view-pooling, a likelihood-based importance weight is generated and used for each viewpoint to take into account the context and reduced viewpoints from the front end devices.

3.2 Video generation and training

There were no videos to test whether the relevant objects were moving with respect to a fixed set of cameras in this particular setup, so generation of videos was required. The modelnet dataset [18] contains 40 classes, with 80 objects as a training set and 20 as a test set. Of the 40 classes, seven were selected (guitar, keyboard, laptop, monitor, person, radio, and Xbox) to generate the videos. There were 3D files (.off) available for each object. Unity [22], known as a game engine, was used to generate a video. In order to use a 3D CAD model, it was essential to convert .off files to the right format that is compatible with the Unity engine. Unity accepts 3D format .fbx, .dae, .3ds, .dxf, and .obj files, so .off files had to be converted. .off files were converted to .dxf files using the software Meshlab. Twenty test cases of seven classes were converted to .dxf files and exported to Unity.

Although 20 test cases of seven classes were converted, 10 were not usable as some were distorted in the Unity setting when converted into 3D files. Therefore, 10 test cases of each class were used to generate the videos. For videos, 12 views were placed as shown in Figure 3.2. Each object was normalized and placed in the center of the 12 sensors. The object took a route from the center of the sensors to 0.6 of the radius from the center. As the object moved from the center to the 0.6 radius at a consistent speed, it took 60 frames. Figure 3.3 shows how the object moved with respect to the frames. The figure shows view0, view3, view7, and view11 as representatives of how the object moved with respect to different views. Since objects have sizes, if the object passed 60 percent of radius, it was not adequate to test the sensors. The design choice of 60 frames
Figure 3.2: Orientation of objects moving in video: (a) shows the 3D orientation of the object and cameras placed, (b) shows orientation facing the front of the object and (c) shows orientation on top of the object. The bottom of the circle is denoted as view0 and from there, counter-clockwise, view1 through view11 was made because security cameras usually have 7.5-15 frames every second. This means that 60 frames contain 4-8 seconds of videos, which is sufficient to see that the object is moving and to test the network.

A selected class of modelnet dataset [18] has been used to train the network. This dataset contains 40 classes, each class containing 100 objects with 12 viewpoints. These 100 objects are divided into 80 training sets (72 training and eight validation) and 20 testing sets. The training was conducted with the 1, 4, and 12 context group, each trained for about 50 epochs. Seven classes were chosen to generate videos and test the model. The chosen classes were home appliances (guitar, keyboard, laptop, monitor, radio, and Xbox) and persons. For training, one powerful GPU (RTX-2080ti) has been used.
Figure 3.3: Object movement in every frame in the left diagram and images on the right, showing every 10 frames and four different views of the same object moving. This is to illustrate how the object is moving.

Figure 3.4: Sample of the videos. (a), (c), and (e) represent the first frame of the video and (b), (d), and (f) represent the sixtieth frame of different 3D CAD objects.
Chapter 4

Results

4.1 Baseline test

To validate that the multiple view performs better than a single view in accuracy, the network has been trained with a single view, using all 12 views. With a single viewpoint, there is no difference between maximum pooling (maxpooling), average pooling (avgpooling), and Weighted average pooling (Wavg-pooling) [1]. As shown in Table 4.1, when examining the accuracy of a network with 12 independently trained single viewpoint, the accuracy is 69.29 percent maximum, 67.85 percent minimum, and an average of 68.51 percent, with the test set of the same view. The test set of the same view means that, for instance, if training is done with view0 out of 12 (view0 through view11), the test set also is tested with view0. However, if tested with the other 11-view test set, the maximum accuracy obtained was 65.71 percent, with a minimum of 60.71 percent.

<table>
<thead>
<tr>
<th>Single view maximum accuracy (same view test)</th>
<th>Single view minimum accuracy (same view test)</th>
<th>Single view average accuracy (same view test)</th>
<th>Single view maximum accuracy (different view test)</th>
<th>Single view minimum accuracy (different view test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>69.29%</td>
<td>67.85%</td>
<td>68.51%</td>
<td>65.71%</td>
<td>60.71%</td>
</tr>
<tr>
<td>Multiple view accuracy (12 views)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97.86%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Accuracy of single view and multiple view
This means there is a context for each viewpoint. The multiple view accuracy reached 97.86 percent when using 12 views to train the network. Twelve views are used as a test set, which leads to the conclusion that multiple view has an advantage over single view.

The generated video was tested with the single view and multiple view trained networks, and had difference in accuracy. Testing was done on the first frame (frame 0), because every object was placed in the center for the videos, under the same environment as the modelnet test sets [18]. As shown in Table 4.2, when tested, accuracy of the single view trained network was 48.17 percent; for the multi-viewpoint network, the accuracy was 84.28 percent.

<table>
<thead>
<tr>
<th>Accuracy of single view trained network tested on generated dataset</th>
<th>Accuracy of multiple view trained network tested on generated dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>48.17%</td>
<td>84.28%</td>
</tr>
</tbody>
</table>

Table 4.2: Accuracy tested on generated video dataset

The accuracy of the overall video seems low compared with the testing on the modelnet test dataset, which approached 95 percent or above. Since the videos are generated by the author, the illumination and viewpoint settings of the original dataset [18] were unknown. The illumination and angles of the viewpoint settings were set similarly by the author, but there was some discrepancy.

In the multiple-view model, testing was done on different context groups (1, 4, and 12 context groups) with maxpooling, avgpooling, and Wavgpooling for each context group, using the modelnet dataset. As illustrated in Table 4.3, the maxpooling, avgpooling, and Wavgpooling reached

<table>
<thead>
<tr>
<th>Context Group</th>
<th>Max</th>
<th>Avg</th>
<th>Wavg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CG</td>
<td>97.86%</td>
<td>96.43%</td>
<td>97.14%</td>
</tr>
<tr>
<td>4 CG</td>
<td>96.43%</td>
<td>97.86%</td>
<td>98.57%</td>
</tr>
<tr>
<td>12 CG</td>
<td>95.71%</td>
<td>97.14%</td>
<td>96.43%</td>
</tr>
</tbody>
</table>

Table 4.3: Accuracy based on number of context group for maxpooling, avgpooling, and Wavgpooling
95 percent or higher. The accuracy was high because the training set was small, with only seven classes with 560 objects. However, the significance is that Wavgpooling with 4 context group had the highest accuracy, as seen in Choi et al. [1].

### 4.2 Viewpoint reduction on videos

After the network’s baseline was evaluated, the video was used to test the network. Out of three different context groups, grouping with 4 context group had the best accuracy, so it was chosen for testing. For weighted average pooling, threshold testing was applied to determine the possibility of reducing viewpoints. The threshold was applied on entropy-based likelihood estimation to prune irrelevant viewpoints.

![Figure 4.1: Accuracy of Wavgpooling without threshold and with threshold for every frame](image)

As shown in Figure 4.1, the maximum accuracy was 85.71 percent for Wavgpooling without threshold and 85.71 percent for Wavgpooling with threshold. The maximum percentage difference between threshold applied and not applied was 4.26 percent and the lowest was -2.86 percent with an average of 1.69 percent accuracy loss in each frame. It is odd to see the accuracy increase as the threshold is applied in frames 1, 32, and 55, because when the threshold is applied, the accuracy
has to decrease. However, this could be because change in weights in an adjacent context might have impacted the consensus.

Figure 4.2: Every five frames accuracy for Wavgpooling, maxpooling, avgpooling, and Wavgpooling with threshold

As shown in Figure 4.2, all of the pooling methods - Wavgpooling with and without threshold, maxpooling, and average pooling - started at 80 percent and decreased as the frames proceeded. This is because, as the frames proceeded, the object moved away from the center, so some viewpoints were not helpful in conducting inference, which caused the accuracy to drop. However, Wavgpooling with and without threshold stayed consistent after the 34th frame, while maxpooling and average pooling decreased in accuracy. This is because maxpooling picked the maximum and average pooling calculated the average, but Wavgpooling predicted locally, which means the heavily weighted viewpoint impacted the inference at the cloud, resulting in better accuracy after frame 34 than by any other pooling method. Instead of frames, it is more intuitive to determine the distance away from the center of viewpoints, as in Figure 4.3.
A threshold is applied in Wavgpooling to reduce the endpoints and save energy. The threshold can reduce accuracy by a maximum of 4.29 percent. The threshold is established to reduce possible views. However, since the objects are different, with each having a different likelihood estimation, we must calculate entropy values to get the likelihood, and find the context and possibility of reducing each objects viewpoint. However, since 70 objects were tested, the tendency of viewpoint reduction and frames were hard to analyze. Therefore, for convenience, Table 4.4 describes how the view reduction was estimated. For each view, the table represents how many objects were a non-zero value of entropy after applying threshold. For simplicity, the table is presented for every five frames, but the evaluation is done on every frame to see the views reduction tendency. For example, in frame 0, viewpoint 2 3 4 is assumed to be views that are excluded in the context, since less than half of the objects (less or equal to 35) have non-zero value of entropy after applying threshold. Thus, examination is done on every frame, like in Figure 4.4, applying the rule introduced using Table 4.4.
Table 4.4: Each number inside the table indicates the number of objects that has non-zero entropy value after applying threshold.

<table>
<thead>
<tr>
<th>Frame</th>
<th>Views</th>
<th>Selection of views reduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>43 52 57 2 26 9 38 56 60 70 70 70</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>42 49 57 2 26 8 40 56 61 70 70 70</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>41 49 56 2 25 10 36 55 58 70 70 70</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>44 51 56 4 20 11 44 54 57 70 70 70</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>42 51 55 3 15 12 43 54 57 70 70 70</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>41 50 54 4 13 8 42 46 55 70 70 70</td>
<td>3</td>
</tr>
<tr>
<td>29</td>
<td>42 50 51 4 10 11 35 44 55 70 70 70</td>
<td>4</td>
</tr>
<tr>
<td>34</td>
<td>37 47 49 5 9 8 32 38 49 70 70 70</td>
<td>4</td>
</tr>
<tr>
<td>39</td>
<td>32 43 50 5 7 8 30 37 45 70 70 70</td>
<td>5</td>
</tr>
<tr>
<td>44</td>
<td>30 38 47 5 7 5 28 31 36 70 70 70</td>
<td>6</td>
</tr>
<tr>
<td>49</td>
<td>25 36 45 7 6 4 25 32 30 70 70 70</td>
<td>7</td>
</tr>
<tr>
<td>54</td>
<td>19 31 39 7 2 3 24 28 26 70 70 70</td>
<td>8</td>
</tr>
<tr>
<td>59</td>
<td>9 25 39 7 2 3 28 16 27 70 70 70</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure 4.4: Views are reduced when the object is moving away from the center. A total of 70 videos were evaluated and if less than half (35 or less) have non-zero value on those viewpoints, that specific viewpoint is reduced.
Figure 4.4 illustrates that when the object is moving away from the center, and out of the static placement of the cameras, there is a tendency of reducing viewpoints. It is shown to be possible to reduce eight viewpoints, 66.6 percent of the viewpoints established in the 12-view video system. If a more aggressive threshold is applied, it can be reduced up to 10 viewpoints, or 83.3 percent of the cameras can be turned off with losing 11.4 percent accuracy. As seen in this figure, the viewpoint reduction doesn't happen dramatically in every frame; rather, it stays consistent and additional views are reduced. Examining every fixed interval of frames, rather than doing likelihood calculation for every frame is crucial, since likelihood estimation can take up to 54.5 percent more computation in Alexnet [1]. As shown in the figure, the interval could be set up based on the established rule. In this setup, the interval of calculating new likelihood is every five frames because, within five frames, there is only one point where reduction occurred. After calculating the entropy of the 70 test cases, we can see how many values of entropy are non-zero. If it is less or equal to 35, it is chosen as the viewpoint to be terminated for the next four frames. However, this policy has to be proven as tested in Figure 4.5.

![Figure 4.5: The impact of viewpoint reduction on Wavgpooling. In frame 0, and in every fourth and ninth frame, the likelihood is calculated, so the accuracy of Wavg with threshold and Wavg with rules applied is the same](image)
Figure 4.5 illustrates that the tendency of the accuracy still holds even when the viewpoints are completely gone. Some points with Wavgpooling with view reduction perform better than Wavgpooling with just threshold applied. This is because some viewpoints that had negative impact on the inference were totally eliminated, resulting in a better accuracy rate. The maximum increase in accuracy was shown in frames 37 and 38. Some frames had a decrease in accuracy; presumably because some viewpoints that had a positive impact on the inference were turned off due to the policy created, which resulted in a maximum of 7.14 percent decrease in accuracy in frame 2. By establishing the policy, the viewpoint reduction can be applied with the trend of accuracy sustaining.

4.3 Viewpoint reduction on videos with contextual mismatch

In a multiple-camera system, an object cannot be oriented in the perfect orientation; a variant of tests should be conducted. The system was trained with the setup where viewpoints are trained in the number of different context groups and evaluated on a test set suitable for that context group. However, that is not ideal in a real-world scenario, because orientation of the object can differ in any way, and there is a possibility of the object being differently sized and not oriented in the perfect way. For testing, the video was generated by rotating 180° of the original videos. For example, the front view was tested on the back view trained model, and vice versa. In addition, the videos generated for testing in Figure 4.2 were normalized so that the object could be seen perfectly; for these videos, normalization was not done.

<table>
<thead>
<tr>
<th>Accuracy of single view trained network tested on generated dataset</th>
<th>Accuracy of multiple view trained network tested on generated dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.14%</td>
<td>52.86%</td>
</tr>
</tbody>
</table>

Table 4.5: Accuracy tested on generated video dataset without normalization and different context
As shown in Table 4.5, the accuracy of the single-view trained network tested on the generated dataset was 17.14 percent and that of the multiple-view trained network was 52.86 percent. It is 31.02 percent lower than the single view, and 31.42 percent lower in multiple view for the dataset in Table 4.2. Thus, the trained network was affected by the dataset according to how the dataset was created, because the training set for the model was normalized perfectly with no error in placement.

The accuracy of Wavgpooling with threshold and without threshold was examined through all 60 frames to determine the accuracy as the object moved away from the center. As shown in Figure 4.6, the maximum accuracy seen in the diagram was 57.14 percent for Wavgpooling without threshold and 54.29 percent for Wavgpooling with threshold in frame 3. The largest percentage drop in accuracy happened in frame 46, which resulted in a 5.71 percent drop when threshold was applied. However, from this, not much can be identified about how Wavgpooling is better than maxpooling or average pooling, so these two methods were tested on the same dataset, which has a contextual mismatch with the trained model.

![Accuracy of Wavgpooling without threshold and with threshold for every frame](image)

Figure 4.6: Accuracy of Wavgpooling without threshold and with threshold for every frame
Figure 4.7: Every 5 frames accuracy for Wavgpooling, maxpooling, avgpooling and Wavgpooling with threshold

As shown in Figure 4.7, average pooling was steady in the range of 45-50 percent accuracy. This is because all the viewpoints were used and averaged out, so 12 viewpoints had an equal contribution to the recognition, and the accuracy did not vary significantly. In maxpooling, there was a significant drop of accuracy after frame 40, because it took the maximum, which did not account for the context. However, results showed that weighted average pooling with or without threshold performed better than those after frame 30, as in the trend of maxpooling in Figure 4.2 in the normalized and context-matched dataset. Since it found the likelihood of each context and performed pooling based on this likelihood, it clearly showed that objects away from the center can be still recognized well compared with avgpooling and maxpooling. In unrestricted setting, where everything is not ideal, the result still shows that Wavgpooling can be used.
Figure 4.8: Distance from the center and accuracy of different pooling. This is basically the same graph as Figure 4.7, with a different x-axis

The video was generated from objects placed in the center of different cameras moving to particular viewpoints to see the corresponding effect of the network in both the frames and distance. As shown in Figure 4.8, average pooling was not significantly impacted as the object moved away from the center. However, there was an impact on maxpooling, as the object moved away from the center, accuracy decreased, and 10 percent more accuracy was lost than in Wavgpooling when it was 0.6 radius away from the center point of the camera. Wavgpooling has been recorded as the highest even with the threshold, 25 percent away from the center.
Figure 4.9: Views reduce when the object is moving away from the center. A total of 70 videos are evaluated. If less than half use the viewpoints, reduction of that specific viewpoint is done.

For views reduced, the rule established in Table 4.4 was used again. Even though the dataset did not achieve good accuracy overall, it showed that views can be reduced, as shown in Figure 4.9. Instead of doing five frames as previously suggested in Figure 4.4, the calculation of likelihood can be done every 12 frames for this system. The likelihood is calculated if there is a point where likelihood and threshold are applied to reduce viewpoints; that particular viewpoint is turned off to save communication cost, with limited loss in accuracy.
Chapter 5

Conclusions

An MVCNN involves multiple viewpoints and aggregation of these viewpoints for better accuracy. A multi-view system is the norm in many situations, such as houses and malls, where multiple security cameras are needed to detect and record. However, not all cameras are valuable in the system.

In this thesis, videos were analyzed with an MVCNN using weighted average pooling. The results showed that reduction of additional viewpoints in videos could be determined by calculating the likelihood for every fixed interval. In doing so, viewpoints could be reduced by up to an additional 41.67 percent more than the conventional decision of viewpoint reduction while performing better than maxpooling or average pooling. In addition, this thesis examined the possibility of employing the system in real-world scenarios where context might be switched to something other than the expected. In that case, a significant decrease in accuracy was demonstrated, but the system worked better than conventional maxpooling and average pooling.

For future work, dynamic application of additional viewpoint reduction can be applied to current architecture other than the fixed interval. The procedure to properly set the fixed interval for additional calculation procedure to reduce viewpoints is unknown, so if the dynamic rule or application could be applied, this architecture could be applied to a real system. Furthermore, different movements other than linear movement could be tested to set up dynamic applications of additional viewpoint reduction. There might be a possibility for context switching within the system,
since the result showed that when the context does not match the trained network, the accuracy significantly decreases.
Bibliography


Academic Vita
Philip Wootaek Shin
Shinwootaek@hotmail.com | pws5345@psu.edu

EDUCATION:
Pennsylvania State University
Master of Science in Computer Science and Engineering
Graduate Research Assistantship (Fall 2018, Spring 2019)
Graduate Teaching Assistantship (Fall 2017)

Bachelor of Science in Computer Engineering
May 2018
Schreyer Honors College, Magna Cum Laude, Dean’s List: All Semesters

Scholarship
Madden Memorial Scholarship (2015-2016)
Dixon Scholarship in Engineering (2016-2017)
Riegel Family Trustee Scholarship, Honors College (2016-2017)
Schumacher Honors Scholarship, Honors College (2017-2018)
GE Comp for the Future Scholarship Engineering (2017-2018)

Award
The President’s Freshman Award (March 2015)
The President Sparks award (April 2016)
Magna Cum Laude (May 2018)

EXPERIENCE:
Pennsylvania State University, State college PA
Graduate Research Assistant for Microsystem Design Lab
• Currently working on Machine learning, Deep Neural Networks, Video Analysis
• Funded by C-BRIC(Center for brain inspired computing) under Dr. Vijaykrishnan Narayanan

Pennsylvania State University, State college PA
Undergraduate Research Assistant for Microsystem Design Lab
• Worked on Non-Boolean computing(Coupled Oscillator)
• Honors research with Dr. Vijaykrishnan Narayanan, Dr John Sampson

Pennsylvania State University, State College, PA
Graduate Teaching Assistant, CMPSC 360 - Discrete Mathematics for Computer Science
• Hold weekly office hours to help students understand concepts of discrete mathematics and quiz problems
• Make course quiz problems and answer sheets and manage coursework graders to grade the quizzes
• Giving lectures and help students solve the questions in weekly recitation

Pennsylvania State University, State College, PA
Grader, CMPSC 360 - Discrete Mathematics for Computer Science
• Grade quizzes, mid-term & final examinations

Pennsylvania State University, State College, PA
Grader, CMPSC 101 – Introduction to Python
• Grade program assignments, mid-term & final examinations

ACTIVITIES:
Publication – Accepted to 56th Design Automation Conference
• Title: Context-Aware Convolutional Neural Network over Distributed System in Collaborative Computing

C-Bric Annual Review -Poster presentation
• Title: Lightweight CNN Design for Edge Devices in Distributed Vision Intelligent System

PROJECTS:
Master’s Thesis Project
• Title: Context Aware Collaborative Object Recognition For Multi Camera Time Series Data
• Implementation in Python, TensorFlow

Undergraduate Honors Thesis Project
• Title: Coupled Oscillator Based FAST Corner Detection
• Implementation in Matlab

ADDITIONAL INFORMATION:
Technical Skills: C, C++, Java, Python, Verilog, Matlab Tensorflow
Linkedin: https://www.linkedin.com/in/philip-shin-82b28a171/