Mirror detection in videos with unsupervised fine-tuning using optical flow and analysis of optical flow in refinement of segmentation mask

Tanish Surana

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2023

Abstract

Mirror detection in videos presents unique challenges, especially given their prevalence and nuanced visual representations. This study delves into the intricacies of this problem, emphasizing the potential of optical flow as a pivotal feature for discerning mirrors. We also introduced a comprehensive unlabelled dataset tailored for video mirror detection, comprising 4054 videos and over 2 million individual frames. Our dataset is orders of magnitude larger can any existing mirror detection dataset. The study comprises two approaches to improve the existing pretrained models: refinement of the segmentation mask by the pretrained mirror detection models and unsupervised fine-tuning of the pretrained models on our larger dataset. Our analysis spotlighted the role of optical flow in anticipating mask dynamics, with RAFT optical flow demonstrating commendable efficacy on mirrors. However, our exploration with pretrained models like VMD-Net and PMD-Net illuminated a concerning overfitting trend, evident from the stark contrast in IoU scores between trained and unseen datasets. Despite these setbacks, this research offers a novel approach to fine-tune pretrained mirror detection models leveraging optical flow.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics Committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Tanish Surana*)

Acknowledgements

I want to express my sincere gratitude to my mentor, Dr. Hakan Bilen, for his essential advice and steadfast support throughout this study. Their advice has been invaluable in constructing this dissertation.

My deepest gratitude to my friends and coworkers for their support and encouragement. Special appreciation to Sparsh Rawal for his thoughtful discussions and technical support during pressing circumstances. I appreciate my family supporting me and the University of Edinburgh offering a supportive learning atmosphere.

Finally, I would like to thank all the other scholars whose work inspired me.
Thank you for participating in this journey, everyone.

Table of Contents

Chapter 1

Introduction

"Objects in the mirror are closer than they appear"

1.1 Motivation

There are mirrors everywhere around us. In the growing field of computer vision, detecting mirrors is an important task; there are various applications where identifying a mirror can become a matter of safety, such as self-driving cars and robot navigation. [7] mentioned that mirrors and reflective objects are one of the top error-causing factors in self-driving cars for person identification. [38] stated that mirrors pose a risk in computer vision. Even in the growing field of Augmented Reality (AR), scene understanding is essential, and being surrounded by mirrors, detecting it is crucial.

Detecting mirrors has been a challenging task as they lack consistent appearance. The appearance of mirrors depends on their surroundings as they reflect the objects around them. Even humans need more time to identify a mirror; for example, we often confuse the path and the mirrors in a mirror maze.

The detection of mirrors is a specialized task; thus the prominent computer vision datasets: ImageNet[11], CIFAR-100[21], CIFAR-10[21], MS COCO[26], PASCAL VOC[14] and ADE20K[39] either don't have mirrors in them or have some images with mirrors in them but are not curated for the task of mirror detection. Mirrors lack unique features, i.e., they have no consistent patterns or textures compared to other objects, such as cars, plants, and animals, which have features that a machine learning model can learn. Thus, the primary state-of-the-art (SOTA) computer vision models, R-CNN, YOLO(You Only Look Once), and Mask R-CNN, struggle to detect mirrors for object detection and segmentation.

A few works have addressed the mirror detection problem [37, 25, 17, 19, 23, 34]. These have a few limitations: First, the datasets used were small. Mirror detection is a challenging problem, as they lack consistent appearance, and more data is required for the model to generalize. Secondly, they all addressed the problem of detecting mirrors in a single RGB image. Many real-world applications for detecting mirrors are video-based, such as robot navigation, self-driving cars, and augmented reality surrounding understanding. Also, videos contain more temporal information, such as the motion of various objects, relative depth of the surfaces, and missing context present in other frames.

A paper by [24], released only a couple of months ago, addressed the issue of detecting mirrors in videos. They created a new labelled dataset of videos containing mirrors. Creating a supervised dataset is expensive and leads to higher costs or a smaller dataset size, which was the case with [24]. Their dataset was small in size when compared to standard computer vision datasets such as MS COCO[26], PASCAL VOC[14] and ADE20K[39].

This research addresses this issue of small dataset size by creating a new large unlabeled dataset and proposes a method for unsupervised finetuning of pretrained mirror detection models on videos using a primary motion cue: optical flow.

1.2 Project Structure

The structure of this research is as follows: Chapter 2 discusses the challenges faced when detecting mirrors, the previous work, the optical flow and the required background. The datasets used, their preparation, and their feature extracted are discussed in chapter 3. The next chapter is about the preliminary analysis of the optical flow model on mirrors. Chapter 5 contains the methodology and results of the refinement of segmentation masks and unsupervised finetuning. Conclusion, limitations and Future scope are discussed in the final chapter.

1.3 Contributions

- Created a large unlabeled dataset of videos containing mirrors
- Analysed the use of optical flow to refine segmentation mask generated by a mirror detection model

- Built an unsupervised finetuning set-up to refine the existing pretrained models using optical flow.

Chapter 2

Background

2.1 Segmentation or Detection

Object detection is the task of detecting what objects are there in the image or video and where they are. Typically, they include labelling the object class and using a bounding box to determine its location.

Semantic segmentation is labelling each pixel as part of a class. There is no differentiation between multiple instances of objects of the same class. For example, if two dogs are in the image, then both dogs will get the same class label.

Instance segmentation is the combination of object detection and semantic segmentation. Here, each instance of an object is given a separate class label.

For this research, we will be doing semantic segmentation i.e. we have only one class, mirror, and won't be differentiating between multiple mirrors if present, i.e. all mirrors detected will be given the same class label.

2.2 Challenges in detecting mirrors for standard computer vision models

As mentioned, mirrors lack consistent appearance, but there are several factors that make mirror detection difficult for standard computer vision models such as R-CNN, YOLO(You Only Look Once) and Mask R-CNN.

- Reflections can change: Mirrors reflect their environment, as opposed to solid things, which have constant appearances. An image's mirror's content is fully determined by the environment it is in, which might change significantly.

- Lack of Unique Features: Since mirrors are essentially characterised by reflections, they frequently lack distinctive textures or patterns. In contrast, distinguishing characteristics like vehicles, animals, and plants may be learned.
- Ambiguities in the boundary: When the mirror's frame is thin or nonexistent, the border between the mirror and its surroundings can be undetectable. This makes it difficult to determine the exact mirror boundary.
- Challenges with Perspective and Depth: Mirrors cause depth to be reversed. Even if an object is literally on the mirror's surface, its reflection can create the impression that it is deeper, which can cause traditional detection algorithms to malfunction.
- Reflections in mirrors can introduce various illumination situations, making it difficult for standard models to generalise well.
- Lack of Training Data: It can be challenging to build a good model to generalise to different mirror scenarios without a large dataset.
- Occlusions and Partial Views: Mirrors may be partially obscured by other objects or only partially visible due to the angle at which they are being seen. Understanding the entire situation is necessary to identify them in these circumstances.
- Mirrors come in various designs, sizes, and forms in real life. They could be concave, convex, or flat. It can be difficult to identify each of these categories in diverse contexts.
- Confusing Surfaces: Highly polished surfaces, such as some metals or aquatic bodies, can also reflect the environment in a mirror-like manner. Typical models might mistake them for mirrors.
- Absence of Contextual Information: In some cases, understanding the context is more important for identifying a mirror than the mirror itself. Recognising furniture, for instance, could indicate the presence of a wall mirror indoors. Standard models may overlook the mirror if they don't consider this contextual awareness.

Mirror detection is a specialized task; hence the previous work done created new or heavily modified modules in their networks.

Hence, the previous work done [37, 25, 17, 19, 23, 33, 34] on detecting mirrors treated mirror detection as a specialized task and made

2.3 Previous Work

2.3.1 Single image-based models

The issue of mirror segmentation was initially addressed by [37]. They released the novel MirrorNet model, the first deep network to detect mirrors, by leveraging the discontinuity of content inside and outside the mirror. They mainly focused on multi-scale contextual contrasted features between mirror and non-mirror areas. They also created a new supervised dataset, MSD, which had over 4000 images of mirrors. [25] further proposed another dataset, PMD, which had over 5000 diverse mirror images and addressed the issue of simple images with MSD. They also proposed a network PMD-Net that focused on correspondence between objects inside and outside the mirror and a module that works on the mirror's edges.

[28] proposed a new RGBD dataset for mirror detection and used a depth-aware mirror detection method to work on the RGBD dataset. Similarly, [33] also made an RGBD dataset and worked on the depth refinement problem on mirror surfaces. These all are binary relationship methods, where they can often get confused between mirror-like objects such as doors, windows and photo frames. Also, RGBD images are not always available and require specialized hardware. Mirror-yolo [23] used a yolo-v4 architecture to detect mirrors using a bounding box and focused on the real-time performance.

Recently, [19] proposed a transformer-based network (SAT-NET) focusing on symmetry. Their network works on the principle that objects inside the mirror are present outside it and are symmetrical. [17] proposed a novel method which focuses on semantic association in the mirror and the background, where they exploited the fact that mirrors are generally placed near wash washing sinks, dressing tables or on a cupboard. Their method fails when mirrors are not in the commonly placed locations.

2.3.2 Video based model: VMD-Net

A couple of months ago, VMD-Net was proposed by [24], and was the first to address the issue of video mirror detection. The proposed model, VMD-Net, uses a novel dual correspondence (DC) module which focuses on capturing correspondence at both

spatial (intra-frame) and temporal (inter/between the frames) levels. They also created a new dataset, which had 269 videos, which accounted for 14,987 individual frames. They had a backbone of ResNext[36], which captured high and low-level features of the frame. The network took in 3 frames: two consecutive frames and a third random frame. Random frames provided a jump in time and gave context which might not be present in the two consecutive frames. The model had 62 million (M) parameters and is the smallest, which brought them real-time performance.

2.4 Optical Flow

Optical flow, or optic flow, is the apparent movement of objects, edges, and surfaces between consecutive video frames. This motion or movement is caused by relative movement between the objects in the frame and the camera.

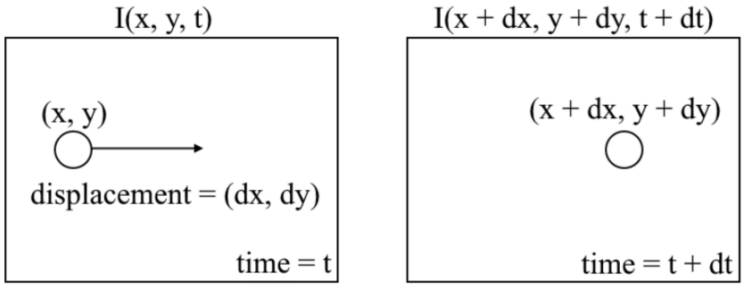


Figure 2.1: The optical flow problem [30]

The Intensity of a pixel can be expressed as I, a function of time t spatial coordinates x and y. The motion or displacement between the frames is represented by (dx, dy) over time dt. So the Intensity at the first frame is $I(x, y, t)$, and if we add the displacement (dx, dy) to it, we get the second frame as $I(x + dx, y + dy, t + dt)$. See figure 2.1.

Essentially, the displacement dx/dt and dy/dt over time (between two consecutive frames) is the motion of a pixel. This motion vector (dx/dt, dy/dt) is optical flow for that specific pixel, which is what various methods try to estimate.

2.4.1 Types of optical flow

There are two types of optical flow estimation: Sparse and Dense optical flow.

2.4.1.1 Sparse Optical Flow

Motion vectors are calculated for specific objects or features in the frame, i.e., not for each pixel in the image. For example, specific objects can be cars or people in the frame. The main advantage of sparse optical flow is that it is more efficient than dense optical flow, as motion is calculated only for specific pixels. For this research, the location, shape and size of the mirrors in the frames is unknown; thus, this research will use Dense Optical Flow, which estimates the optical flow for all the pixels, as explained in section 2.3.1.2.

2.4.1.2 Dense Optical Flow

In dense optical flow, motion is calculated for each pixel in the frame, which describes the movement of all pixels between two consecutive frames. These methods are more computationally expensive as they require calculating motion for all pixels. The primary applications are video stabilisation and video compression.

Over the years, many dense optical flow estimation methods have been proposed. Classical methods such as Horn and Schunck (HS) [18] and Lucas-Kanade (LK) [27] use the Talyer series expansion and brightness constancy assumption, solving for the flow parameters using a partial derivative.

With the rise of deep learning, better estimation methods were proposed recently. FlowNet and FlowNet 2.0 [13] were the early deep learning models using Convolutional Neural Networks (CNN), whereas the later version incorporated multiple FlowNets to improve its accuracy. These are supervised models. PWC-Net [32] provided improved accuracy and lower computation cost over FlowNet and used a pyramid, warping and cost volume structure.

Recurrent All-Pairs Field Transforms (RAFT) [35] is another deep learning-based optical flow estimation method. It uses a recurrent neural network backbone and all-pair comparison to estimate the flow iteratively. The recurrent part of the model iteratively improves the estimations, and the all-pair comparison computes the correlation between pixels of the local neighbourhood for each pixel in both frames. RAFT combines supervised and self-supervised learning and has state-of-the-art performance across multiple benchmarks, including Sintel [9] and KITTI [15].

In this research, we will use the RAFT model over other deep learning methods and other methods for estimating dense optical flow for the following reasons. First, the model has state-of-the-art performance in terms of accuracy. Second, the model

has real-time computation capabilities. Hence, it can be used in our pipeline to detect mirrors without adding computational overheads. Thirdly, and most importantly, the RAFT model has excellent performances over various datasets. Thus, it generalises well and can be used for the datasets used in this research.

2.4.2 Representation of optical flow

In Dense optical flow, each pixel has its own motion vector. Each pixel in the image has a vector that denotes the movement's amplitude and direction. The magnitude denotes the rate of change in a pixel's location between the first and second frames, and the direction indicates that change.

Figure 2.2: Color Wheel used for representing optical flow [10]

Using a colour wheel (shown in the figure 2.2), where each direction is assigned a colour, is one of the standard techniques to visualise the optical flow. It's common to employ the Hue, Saturation, and Value (HSV) colour space. The direction of motion is conveyed by hue. The magnitude is encoded by the value (or brightness). For instance, a movement to the right may be shown in red, a movement upward could be highlighted in green, and faster movements would be highlighted in more light. For example, see the optical flow image generated in figure 2.6.

2.4.3 Why can optical flow be useful for detecting mirrors?

The principle of motion parallax [8] states that objects closer to a viewpoint have a faster perceived motion when compared to objects further away, see figure 2.3. The objects inside the mirrors (reflections) have a greater depth when compared to the immediate surface around the mirror. Thus, a video of a mirror with camera movement will have different motions of objects/pixels (optical flow) for objects inside the mirror than the surrounding background.

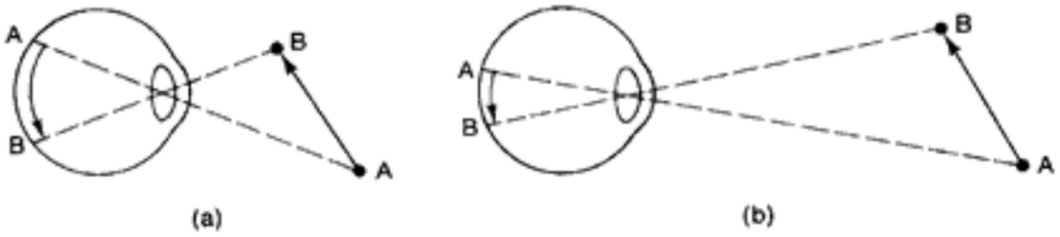


Figure 2.3: The movement A to B is the same in both (a) and (b), but due to the increased distance of movement in (b), the speed of movement in the retina is slower in (b) than in (a)[29]. Similarly, this effect occurs in the camera taking a video

This means that when a camera moves in a parallel plane to the mirror surface, the objects inside the mirror will have a slower motion when compared to the immediate surface surrounding the mirror and the mirror frame. Even if the background surrounding the mirror is far away(having greater depth), the mirror's frame will have a relatively different motion when compared to the objects inside the mirror. Thus creating a boundary in the optical flow image, see figure 2.6. Thus, an optical flow image may have a distinct boundary on the edges of the mirror. Hence, optical flow can be a useful feature for models detecting mirrors.

Figure 2.4: The leftmost image is the optical flow image generated by the RAFT model; the middle two images are random consecutive frames, and the rightmost image is the ground truth mask showing the segmentation mask for the mirror. The optical flow image (leftmost image) shows there is a clear boundary between the motion of objects inside the mirror and its immediate surroundings

2.5 Segmentation models

These are the segmentation models that we used in the refinement of our segmentation mask.

2.5.1 Convolutional Autoencoder

A Convolutional Autoencoder (CAE) is an autoencoder that uses convolution, instead of a fully connected neural network found in an autoencoder. These are designed to handle images, as they use convolution and deconvolutional blocks which preserves the spatial information in the image.

Autoencoders have two parts, encoder and decoder. The encoder compresses the input into a representation of latent space. It reduces the dimensionality of the input image and encodes it as an internal fixed-size representation. Decoder reconstructs the input data from the latent space representation. CAE use convolutional and deconvolutional block in the encoder and decoder.

CAE are primarily designed for feature extraction and dimensionality reduction and applications such as noise removal and anomaly detection.

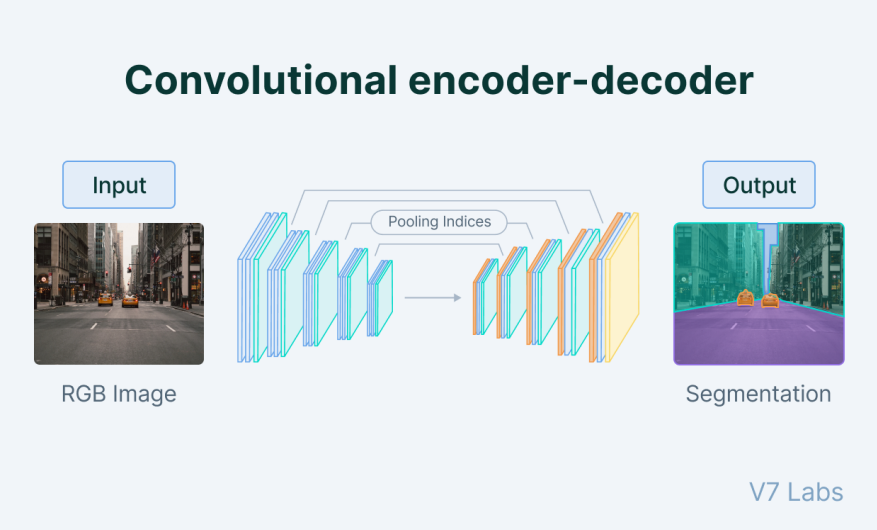


Figure 2.5: Convolutional Autoencoder architecture [22]

2.5.2 U-Net

Like CAE, U-net has an encoder (downsampling path) and decoder (upsampling path) that use convolutional and deconvolutional blocks. But, it has skip connections between the encoder and decoder blocks. This makes it possible to combine low-level and high-level characteristics during upsampling, improving the segmentation's quality.

U-Nets ared designed for image segmentation tasks, where it was first introduced for biomedical semantic segmentation [31].

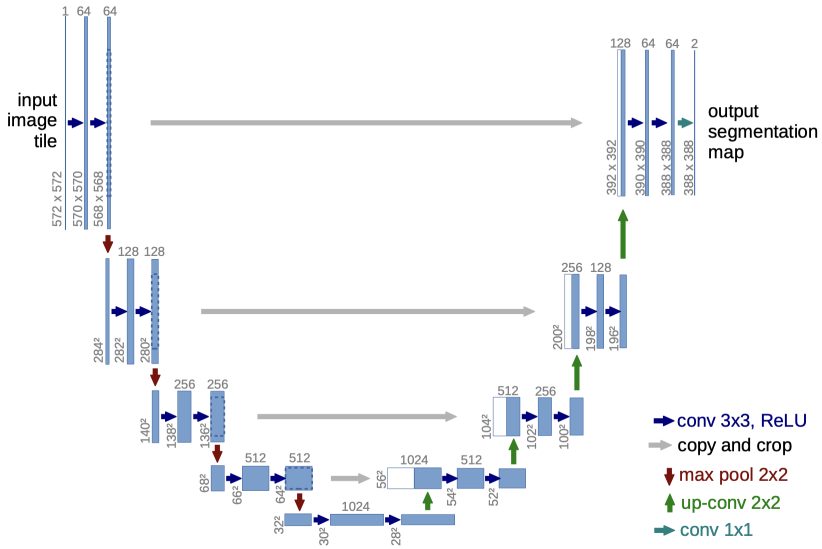


Figure 2.6: U-net architecture [31]

Chapter 3

Dataset

For our project, we used two datasets, first, the VMD dataset, a supervised dataset and second, the unlabeled dataset created.

3.1 Existing Datasets

Since only one work has been done on mirror detection in videos, there exists only one dataset for mirror detection in videos, Video Mirror Detection (VMD) dataset, which was introduced by [24]. The dataset consists of 269 videos, which have 14,987 frames in total. This is a supervised dataset, i.e., a ground truth mask for each image is available. The total video length is about 500 seconds, and each frame is set to a resolution of 1280x720 pixels. Most videos have about 60 frames, i.e., the average length for each video is about 2 seconds.

3.2 Unsupervised dataset created

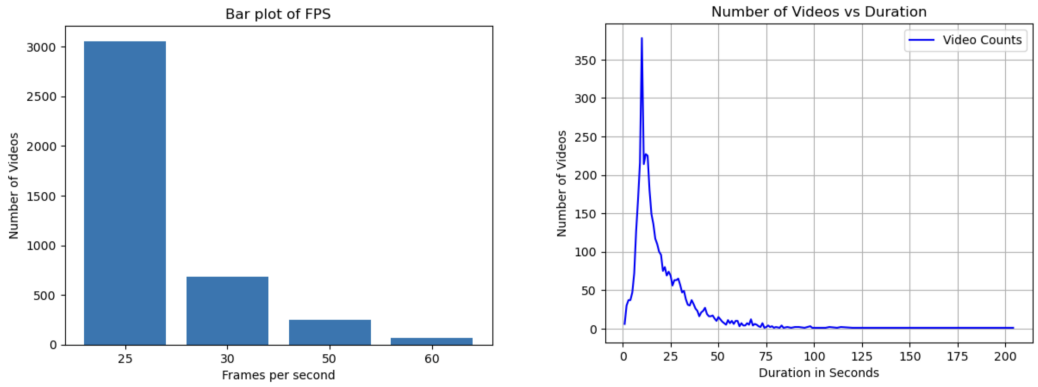
Creating a labelled dataset requires manual annotation, i.e. making a mask for each image, which is expensive and requires large amounts of human labour. The VMD dataset has 14987 individual frames with a ground truth mask made for each frame manually. VMD dataset is larger when compared to the previous MSD and PMD dataset, which has about 4,018 and 6,461 supervised frames, respectively.

ImageNet[12], a popular object recognition dataset has about 1.2 millions images for training and 50,000 for testing. Similarly, COCO[1], a large-scale object detection and segmentation dataset, has 328,000 images. So, having 14,987 images in VMD

dataset means it is a relatively smaller dataset, and for detecting mirrors, which is a difficult task to generalise, increasing dataset size would be beneficial.

We created a large unlablled dataset of videos containing mirrors. **Our dataset has 4054 videos and has more than 2 million frames.**

The dataset we gathered is publicly accessible with a royalty-free licence and under Creative Commons Zero (CC0) licence [3, 4], and it was obtained from Pexels.com [2], a free stock photo and video website where photographs and films can be used for all personal and professional reasons, including distribution, download, copying, and modification, without the need for any credit or copyright.



(a) (b)

Figure 3.1: (a) Distribution of Frames per second (FPS), (b) Distribution of Duration of videos

Pexels's api [5] was used to get all videos from the website with the search term "Mirrors", this resulted in 4054 videos in total. The quality for videos was 1280x720 (or 720x1280 if portrait), and if the quality wasn't available, higher quality was downloaded and later scaled down to 720p resolution.

The statistics for our dataset are shown in the figure and table 3.1. The total size of the created dataset is 35.5 GB.

3.2.0.1 Data Preparation

To convert the dataset into frames, first, all the 50 frames per second (FPS) videos were converted to 25 FPS; then, all the 60 fps videos were converted to 30 fps by skipping every alternate frame. In general, standard FPS are 24, 30, 50 and 60. But to make the dataset generalized, two categories were made: 25 and 30 FPS. *Note: 25 fps cannot be*

	Total	Portrait	Landscape
Number of Videos	4054	1392	2663
Average FPS	27.875	27.227	28.205
Average Duration (seconds)	20.274	18.938	20.974
Total Duration	82174	26362	55812

Table 3.1: Statistical Analysis of the dataset created

converted to 24fps without introducing missing frames, and 25 is close enough to 24, so downstream models won't get affected.

Then, all the portrait videos were rotated to make them landscaped. The aspect ratio was set to 16:9; if videos didn't have this ratio, they were cropped in. After this, all the frames were scaled to a width of 1280 pixels and height of 720 pixels, which resulted in over 2 million individual frames.

3.2.0.2 Limitation of the Dataset:

The dataset is from a stock photos and videos website, where the content is used for high-quality advertisement, b-roll images, etc. The quality of the videos was very high and of professional grade. In common applications of detecting mirrors such as robot navigation, Augmented reality scene understanding and self-driving cars, the camera quality will differ a lot. Also, in our dataset, the videos have background blur present in some cases, which might confuse the downstream models for detecting mirrors. Lastly, these videos have a slower motion when compared to any other real-life videos, which could also affect any pretrained model which was trained on some other dataset.

Chapter 4

Preliminary Analysis

4.1 RAFT Optical flow analysis on mirrors

The following experiments were done to analyze the performance of RAFT optical flow estimation on mirrors. This experiment aimed to check if RAFT optical flow can be used to predict the movement of mirrors between consecutive frames.

Hypothesis: The optical flow vectors calculated between two consecutive frames, when added to the ground truth segmentation mask of the first frame, will give the mask at the second frame.

$$\text{mask}_{t+1}^{\text{predicted}} = \text{optical\_flow}(\text{frame}_t, \text{frame}_{t+1}) + \text{mask}_t^{\text{gt}} \tag{4.1}$$

$$\text{mask}_{t+1}^{\text{predicted}} \approx \text{mask}_{t+1}^{\text{gt}} \tag{4.2}$$

The dataset used for this analysis was the VMD-dataset, where all 14987 frames where used.

Evaluation Metric: The Jaccard Index, also known as intersection over union (IoU), is a statistic used to assess how similar two binary masks are to one another. It's frequently used in computer vision binary segmentation tasks to gauge how well the predicted binary mask overlaps with the actual mask.

Below is the formula for IoU:

$$\text{IoU} = \frac{\text{Intersection of ground truth and predicted masks}}{\text{Union of ground truth and predicted masks}} \tag{4.3}$$

4.1.1 Benchmark: No optical flow

The optical flow's magnitude will be small since there is little movement between two consecutive frames. If the optical flow estimate performs poorly, adding an inaccurate optical flow to the mask at the first frame will still give us a high IoU. i.e. there is still a higher initial overlap between the two masks of the frames, and moving a mask slightly will still result in high IoU.

To solve this issue, the overlap between the first and second masks is used as a benchmark, which means no optical flow is added. This will give us a base overlap, and if the correct optical flow is estimated and added in eq 4.1, the predicted mask and ground truth mask at the second frame should have a higher IoU.

Result: Average IoU 0.957. This will become the benchmark for the experiments below; if IoU is lower, it could indicate that optical flow is not working well.

4.1.2 Adding optical flow vectors

Here, the optical flow estimated was added to mask_t to get $\text{mask}_{t+1}^{\text{predicted}}$ as in the equation 4.1.

Result: Average IoU: 0.921. After investigating the predicted mask generated, it was found that the masks had cracks caused by individual pixels in the masks having different directions of optical flow vectors.

4.1.3 Average optical flow

To resolve the issues of cracks caused by different direction of optical flow vectors, an average is taken. Thus all the pixels inside the mask will have the same vector direction and magnitude, and the average optical flow should tell us the movement of the mask.

Result: Average IoU: 0.934, this still is lower than the benchmark of 0.95, but the problem of cracks was removed and had a higher IoU than using optical flow without averaging them.

The boxplot in figure 4.1 shows that there are videos that have very poor performance. After analysing the outliers, it was found that it was being caused by three major factors.

- First, the videos that had IoU lower than 0.8 had parts of mirrors out of the frame. So, as the camera is moved, the mirror's area increases (part of mirror that is outside the frame is added). The method used in the above experiment didn't

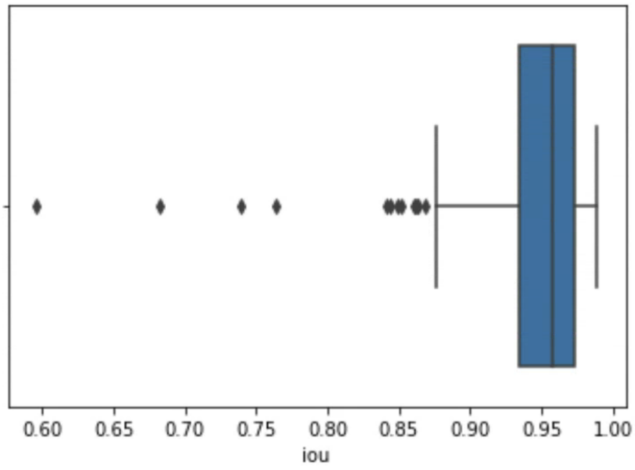


Figure 4.1: Boxplot of IoU generated for all videos in the VMD dataset

account for this, so the predicted mask size remains the same as the mask at frame 1. Hence degrading the performance.

- Secondly, the rest of the outliers were caused by an object (in front of the mirror) or camera moving towards or away from the mirror, thus changing the size of the mirror, but in the above method, the size of the predicted mask is not changed, hence resulting in poor performance.

In both cases, the size of the segmentation mask was changing, which caused the lower IoU scores.

- Also, when an object is present in front of the mirror, or a relatively huge object w.r.t the mirror has a lot of motion, then averaging the optical flow captures the movement of the object rather than the motion of the mirror, then resulting in a wrong estimation of optical flow.

This is not caused by the poor performance of the RAFT optical flow model but due to the experimentation methodology of adding average optical flow.

All these lower IoU scores are due to the limitations and drawbacks of the methodology selected; thus, removing the outliers doesn't affect the aim of the experiment. After removing the outliers, the **mean IoU was 0.968**, which is higher than the benchmark, thus indicating that the optical flow estimated by RAFT was working well in predicting the camera's movement even if mirrors are present in the video.

Table 4.1: Experiment Results

Experiment name	**IoU**
Benchmark	0.957
Optical flow	0.921
Average optical flow	0.934
Average optical flow without outliers	**0.968**

4.1.4 Manual Inspection

We also manually inspected the optical flow estimated for many of the frames and found that the optical flow worked as expected. The objects inside the mirror had a different flow than those outside the mirror, which is the same as what was discussed in section 2.3.3. In conclusion, after considering the experiments and manual inspection, RAFT optical flow showed no cases where it failed or predicted wrong estimates.

4.1.5 Cases where optical flow might not be a good feature

- **Static Scenes**: When there is no camera movement or no motion between consecutive frames, the optical flow might not give any meaningful information for the downstream task.
- **Rapid Scenes**: When there is a fast movement, it might confuse the optical flow estimation model.

Chapter 5

Methodology and Results

This chapter contains two sets of experiments; the first is refining the segmentation mask generated by a mirror detection model using optical flow as a feature. The second is about the unsupervised finetuning of a pretrained mirror detection model using optical flow.

5.1 Loss Function

For binary segmentation, the standard loss function is binary cross entropy (BCE). For every pixel, we compute the log loss between the ground truth label (1 or 0) and the probability prediction p.

$$\text{BCE} = -\left(y\log(p) + (1-y)\log(1-p)\right) \tag{5.1}$$

[6], introduce a new loss function, Lovasz-Softmax loss function, which prioritises improving the IoU that the standard BCE.

For the unsupervised fine-tuning, the Lovasz-softmax loss function (equation 5.2) was adopted, as our pretrained model, VMD-Net, was trained using it, [24]. Where L_h is the hinge loss, P_i, G_i and M_i are intermediate output, ground truth and final segmentation mask from the VMD-Net, respectively.

$$L = \sum_{i \in \{t, t+1, n\}} \left(L_h(P_i, G_i) + L_h(M_i, G_i)\right) \tag{5.2}$$

We experiment with the Lovasz-softmax loss function and BCE for the refinement experiments. For Lovasz softmax loss, only the hinge loss is considered between ground truth and the final mask of the refinement module.

$$L = L_h(M, G) \tag{5.3}$$

5.2 Evaluation Metrics

Since the release of VMD-dataset [24], supervised labels are available; hence evaluation can be done on this using the standard evaluation metrics used in supervised binary segmentation, which are Intersection over Union (IoU), Mean absolute error (MAE), and pixel accuracy.

5.3 Segmentation mask refinement using optical flow

This study focuses on using the optical flow features to refine the segmentation mask made by the pretrained VMD-Net model. This experiment aims to see if the optical flow can improve the segmentation mask made by a pretrained mirror detection model.

We experimented with various types of optical flow features, initially, it was just the optical flow image (colour wheel representation), then a segmented optical flow image and finally with an edge detector on the optical flow image. We also tested out two refinement models, a Convolutional Autoencoders and a U-Net.

Hypothesis: Optical flow estimates having distinct flow vectors between the inside and outside of the mirror can be a useful feature for the refinement model and can learn to use the distinction to improve the segmentation mask. See ground truth image and optical flow image in figure 5.1

5.3.1 Dataset

The dataset used here is the VMD dataset, which has labels, i.e. ground truth images. The dataset has about 14,500 images and was split into train, test and validation. The train set was kept the same as the original VMD dataset. The test set of the original VMD dataset was further split into test and validation, with around 3,000 images each.

5.3.2 Optical Flow features

These are the various features that were used in the following experiments.

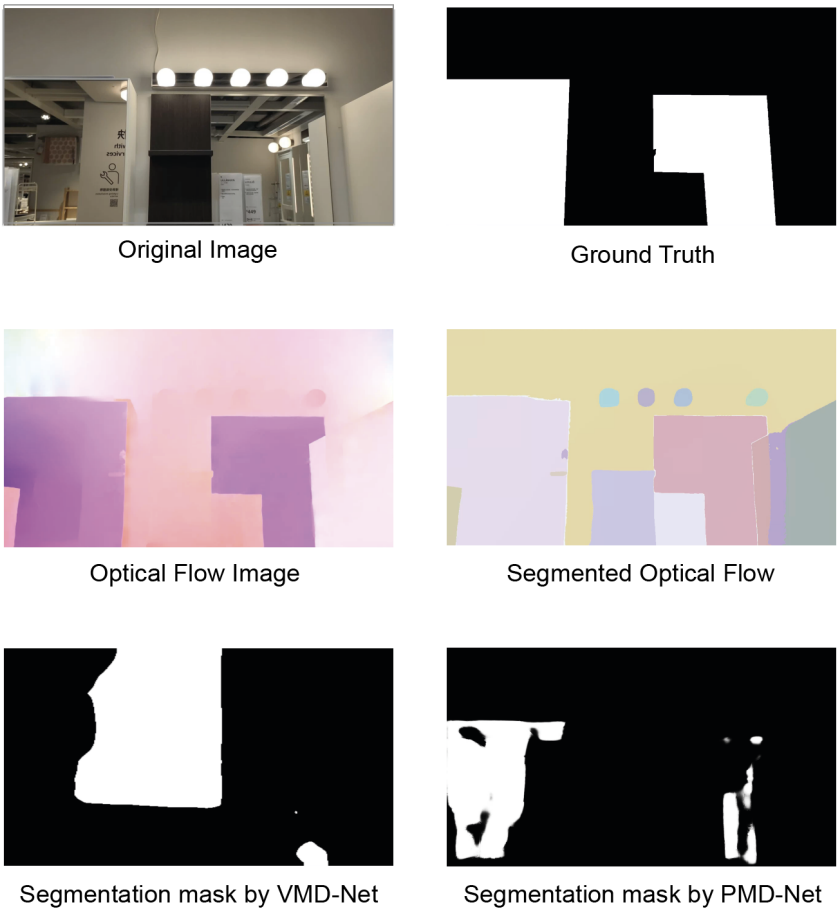


Figure 5.1: Sample of images and features used in the refinement experiment

5.3.2.1 Colour Wheel representation of Optical Flow

Using the RAFT optical flow estimate model, Optical Flow was estimated for each consecutive frame in the VMD dataset. This optical flow was then converted to the colour wheel representation of the optical flow. See the optical flow image in Figure 5.1

5.3.2.2 Segmented Optical Flow features

The optical flow images (Colour wheel representation) had a wide range of colours. So many shades of colour can overwhelm the refinement model. To reduce this, clustering the patches having similar optical flow can reduce the number of colours and essentially group all the objects or sub-objects having similar optical flow.

To cluster or combine the areas having similar optical flow, Segment Anything Model (SAM) [20] was used. SAM is powerful zero-shot generalisation to unknown objects and images, thus no further training is required for the purpose of clustering

similar optical flow areas.

Here the (vit-h) SAM model [20] was used, with points per side set to 8. The input for the SAM model was optical flow images generated in the above section. See the Segmented optical flow image in Figure 5.1. The number of colours has been reduced drastically and only the useful boundaries are kept. Compare it to the optical flow image in Figure 5.1.

5.3.2.3 Edge detection of Optical Flow images

The optical flow image and the segmented optical flow images, still have a problem, the colours of the objects are not the same in different frames. See figure 5.2. The colour in an optical flow image depends on the motion of the objects and the colour in segmented optical flow is randomly allocated as it was a zero-shot run. This can obscure the refinement model from learning, as it could associate colour with objects. We will call this problem the *Colour Problem*. To solve this, we applied a Sobel edge detector [16] to both of these features before sending it to the refinement model. This removes all the colours and keeps in only the shape of the objects.

All the images were first converted to grayscale before sending it into the Sobel edge detector. Sobel uses a simple 3x3 convolution kernel and hence it is computationally fast. Also, sobel might miss finer edges and can have issues when there is noise in the images, but for our purpose the images had clear boundaries and had very little noise.

5.3.3 Methodology, Experiments and Results

To refine the segmentation mask, we tested two types of model, a Convolutional Autoencoder (CAE) and a U-Net model; let's call them refinement models. The refinement model takes input as an image and outputs a binary segmentation mask.

The optical flow feature and rough segmentation mask generated by the pretrained VMD-Net model are of the same size (Width 1280 and height 720). The depth of the optical flow feature depends on the feature used, the depth of the rough segmentation mask is one. The optical flow feature and rough segmentation mask are concatinated together and then is given as the input to the refinement model. The architecture of this setup can be seen in the figure 5.3.

In the field of image processing, Convolutional Autoencoders (CAEs) and U-Nets are both common architectures, particularly for segmentation tasks. CAEs are faster to train and due to the encoder-decoder structure, CAEs are proficient at learning hierarchi-

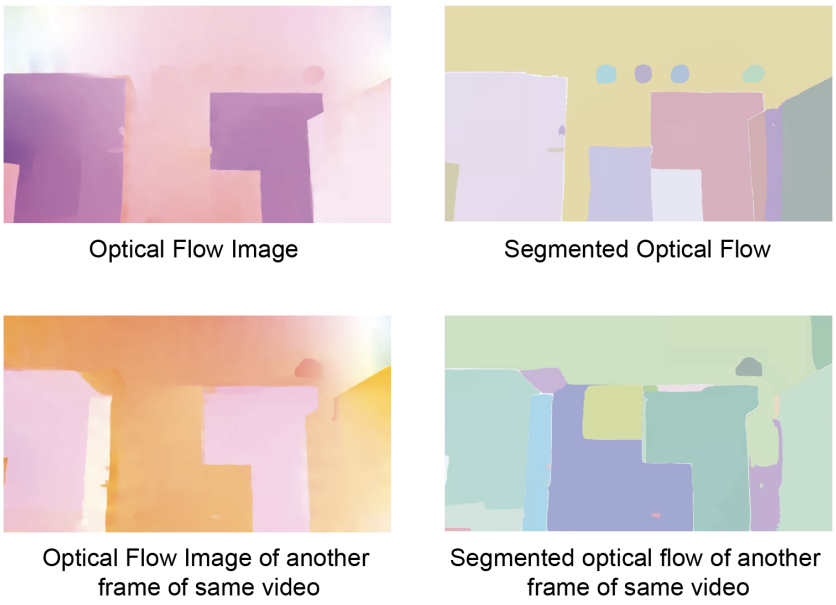


Figure 5.2: Optical flow and Segmented optical flow two different frames of the same video. It shows different colour assigned to same objects in the images

cal features from the data. But, CAEs are not by nature segmentation-designed, so we also used U-Nets. U-Net is naturally suited for binary segmentation problems because it was specifically created for biomedical image segmentation. Better localization in the segmentation maps is made possible by the skip connections in U-Net, which aid in maintaining the spatial context.

The model architecture can be found in appendix.

Experimental Setup: The input and output image sizes were lower to 384x384, as the VMD-Net model had the same size inputs and outputs. Supervised training was used with early stopping, and with Adam optimizer with a learning rate set to 1e-3, with a batch size of 16 and a dropout rate of (0.2, 0.3, 0.5). The model was trained on a GTX 1660 6GB GPU.

The **benchmark** for this experiment is the pretrained VMD-Net model which has an IoU of 0.56 on the test set.

Now we will discuss the experiments done, their results and discussion.

5.3.3.1 Baseline: Only Segmentation mask from VMD-Net model

In this experiment, the refinement model (CAE) is only given the segmentation mask from the pretrained model. This is to check the overall performance of the refinement model and to see how much information is lost in the encoder-decoder step of the

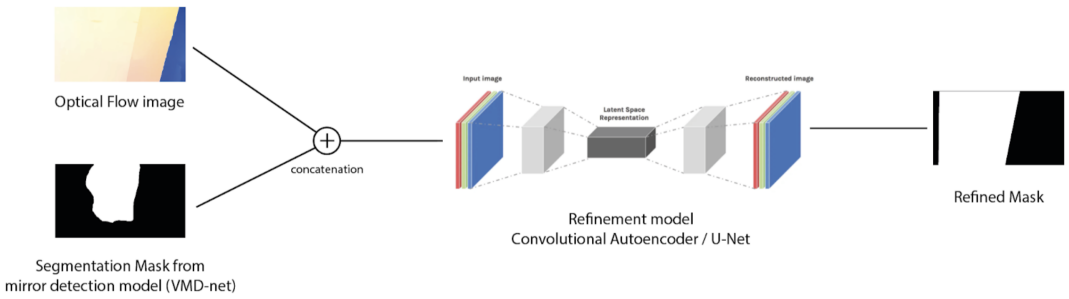


Figure 5.3: Architecture of the refinement process. The input is optical flow image and segmentation mask, which is concatinated and pass into the refinement model (Convolution Autoencoder (CAE) or U-Net (note the figure has a CAE)) which generates the final refined mask.

model.

Results and discussion: BCE was used as the loss function, with that 0.51 IoU was achieved on the test set. This is lower than 0.56 IoU of the benchmark. With Lovasz softmax loss equation (5.3), IoU on the test set was 0.545. This is still slightly lower than the benchmark, but this loss can be due to the information loss in the encoder-decoder step.

Similarly, we experimented this on the U-net architecture and the IoU for the test set was 0.546. For future experiments, we opted for the Lovasz softmax loss function.

5.3.3.2 Optical Flow only

This experiment checked if optical flow (Colour wheel) alone could detect mirrors using this training architecture. ***Hypothesis:*** The model won't be able to learn where the mirror is, as in the optical flow image has a lot of patches (areas having similar optical flow). So, even if the model is capable of detecting patches as objects, it won't be able to tell which one is a mirror.

Result and discussion: The IoU for this was 0 and MAE was 1, this means our hypothesis was correct, and the model wasn't able to just use the optical flow image

Similarly, we did the same experiment on U-net and with segmented optical flow features and edge-detected optical flow features. But the result was the same.

5.3.3.3 Optical Flow and segmentation mask

In this experiment, both the optical flow image and segmentation mask from the pretrained model, VMD-Net, were given as input to the refinement model. ***Hypothesis:*** there should be a significant improvement as the model can learn to use the optical features.

Results and discussion: With both the refinement model, the IoU on the test set was 0.544, which is similar to the baseline. Some improvement was expected, but it could be that the model got overwhelmed by large amounts of colour in the optical flow image.

To check this we passed in the segmented optical flow features but got a similar IoU of 0.545. This means the model is not using the optical flow, as the result was the same as just passing in only the segmentation mask. We did the same experiment with grayscale images of both the features but got the same result. Even the grayscale images have different gray values problem so this is the same as the Colour problem mentioned in section (5.3.2.3).

5.3.3.4 Edge Detected optical flow features and Segmentation mask

In this experiment, we used the edge detected optical flow image instead of the normal colour wheel optical flow image. ***Hypothesis:*** Any issues which were caused by the colour problem, should not occur in this experiment as we are using edge-detected features. The edges still preserve the shape of the patches and objects in the optical flow image.

Results and Discussion: Again the results were the same, no improvement in IoU and other metrics, for both the models (CAE and U-net). Even with varying the hyperparameters: dropout probability, learning rate and batch size, there was no improvement.

5.3.4 Discussion

The models weren't able to use the optical flow features. This could be due to the following:

- **Optical Flow not present in many frames:** As mentioned in section 4.1.5, if there is no motion between consecutive frames, the optical flow will not give any useful information. This was observed in our optical flow features created, where

a set of 2-3 frames had no optical flow. This means while training, the model will only use the segmentation mask from the pretrained VMD-Net model.

- **RAFT model**: Another cause could be if the RAFT model didn't give good optical flow estimates. But this wasn't observed after manually checking the features. Almost all of the frames had clear separation in the boundaries of mirrors and other objects.

- **Low data size:** It might be possible that there is not enough data for the models to learn the optical flow images.

- **VMD-Net:** We observed that the training IoU scores of the vanilla VMD-Net was over 0.9, and the testing IoU was 0.56. This was also the case with the above experiments. After analysing the segmentation masks generated by the VMD-Net model, we observed that the training masks were near perfect in almost all of the videos, but in the testing set, the masks generated by the VMD-Net were of poor quality. . In the test set, there were many cases where the masks generated were blank (no mirror detected) and had completely wrong masks, i.e. detected some other object instead of mirrors and cases where a partial part of the mirror was detected. This means the model wasn't able to generalize well. For example, see the segmentation mask by VMD-Net in figure **??**.

The poor performance of VMD-Net in test cases and near perfect mask in the training set meant that in the above experiments, the model was just using the segmentation mask to recreate the same segmentation mask using the encoder-decoder part and was completely ignoring the optical flow features, This gave the refinement model good mask when it was training, but while testing since the segmentation mask from VMD-Net were of poor quality, it again only used the segmentation mask and ignored the optical flow features resulting in poor performance.

To resolve this issue of poor testing mask and good training mask, few more experiments were conducted.

1. First, to equalize the quality of the segmentation mask by VMD-Net, we added missing data to the training mask generated. Two experiments were conducted. First, constant missing data, where 50% of the pixels were randomly selected and were given a random pixel value. Second, as the epochs increase, we reduce the missing data per cent from 50 to 0 in a linear way. Similarly it was also done with

30 % as the starting missing data. In all of the experiment the results dropped, reaching an IoU of only 0.356 in testing.

2. Next, we only used the testing data, so only poor quality mask will be given as input. The validation set was used as training data, and the test set remained the same. After testing all types of features, the performance didn't improve. This could be due to the poor quality of the VMD-Net mask; the model didn't learn which part of optical flow features the model should use.

3. Lastly, we checked another mirror detection model, PMD-Net [25]. We did inference on the VMD data using the pretrained PMD-Net, the mask generated were again of poor quality, but in both test and training data. This was due to the fact for PMD-Net both of these were unseen data.

5.3.5 Conclusion

The above methodology couldn't use optical flow features to improve the segmentation mask. This was mainly due to the overfitting nature of IoU in both VMD-Net and PMD-Net models, where they both performed poorly on unseen data. Also, one cannot conclude if optical flow features are useful or useless to refine the segmentation mask generated by a pretrained model, as the pretrained model was the bottleneck here. Other issues like static frames, and " colour problem" of optical flow features and the performance of RAFT on detecting mirrors could not be addressed now.

5.4 Unsupervised Fine-tuning using Optical Flow

This experiment aims to improve the VMD-Net mirror detection model by finetuning it using the larger unlabelled dataset.

As aforementioned, mirror detection is challenging, and standard computer vision models might find it difficult to detect the mirrors. Henceforth, all the previous work which addresses the issue of mirror detection, developed a specialized model for it. VMD-Net was the only model to address the issue of mirror detection in videos, and when compared to existing single-image mirror detection models, it had the best performance.

Due to being limited by the timeframe, it was decided to use pretrained mirror detection VMD-Net[24] for finetuning it with our larger unlabelled dataset to improve

its performance.

5.4.1 Dataset

We will use the unsupervised dataset that we created, which had 4054 videos. We will be using the landscape videos and only using the first 60 frames of the videos. The average length of the videos is around 20 seconds, and each video will have 500 frames. Since all the frames will be similar, feeding the model a lot of similar frames could prevent the model from generalizing. Also, it will make the training epochs faster.

We only use the landscape videos as we have a lot of data. We have over 2,663 videos, which results in over 159,000 total frames.

For testing, the test set of the VMD dataset was used.

5.4.1.1 Analysis of VMD-Net on our dataset

As mentioned in the Refinement analysis section, the VMD-Net model performed poorly on unseen data compared to the seen data. When inferred on our dataset, it had similar results, where in some videos it was working well (figure A.1 in appendix), and in some videos it detected the mirror but with some extra objects see figure A.2 and in some it showed completely blank mask A.3.

5.4.2 Methodology

Hypothesis: The two consecutive masks generated by the VMD-Net should have the same optical flow as the optical flow between the two consecutive frames.

$$\text{optical\_flow}(mask_t, mask_{t+1}) \approx \text{optical\_flow}(frame_t, frame_{t+1}) \tag{5.4}$$

We can add an optical flow estimate between two consecutive frames to the first mask generated by VMD-Net to get the second mask. We can predict the second mask using the optical flow between frames and the first mask.

$$\text{mask}_{t+1}^{\text{predicted}} = \text{Optical\_flow}(frame_t, frame_{t+1}) + mask_t \tag{5.5}$$

This predicted mask at time t+1 should be similar or identical to the mask generated by VMD-Net at time t+1

$$\text{mask}_{t+1}^{\text{predicted}} \approx \text{mask}_{t+1} \tag{5.6}$$

A single training epoch is divided into two stages: creating new labelled data from unlabelled data and updating weights of the VMD-Net on the newly labelled data.

5.4.2.1 Creating new labelled data from unlabelled data

The VMD-Net takes in input as three frames. The two consecutive frames: $frame_t$, $frame_{t+1}$ and a random frame, $frame_n$, from the same video. The VMD-Net outputs three masks for each of the frames, $mask_t$, $mask_{t+1}$ and $mask_n$.

The optical flow estimated by RAFT between the two consecutive frames is added to the $mask_t$ generated by VMD-Net; this gives us $mask_{t+1}^{\text{predicted}}$. Then IoU is calculated, between the $mask_{t+1}^{\text{predicted}}$ and $mask_{t+1}$ by VMD-Net.

This is done for all the frames in our unlabelled dataset, and all the IoU and (frame, mask) pairs are stored temporarily. Then, the top ”n” IoU pairs are selected, added into a new labelled dataset, and removed from the unlabelled dataset (The mask generated by VMD-Net are now treated as ground truth images). This process is explained in the figure 5.4

5.4.2.2 Updating the VMD-Net model

Now, with the new labelled dataset, the VMD-Net model is finetuned, i.e. it is trained on this new labelled dataset. The loss function and the evaluation metrics used are the same as the ones used by VMD-Net.

After this, one training epoch is finished, and the process is repeated.

5.4.3 Experimental setup

Computational Specification: The VMD-Net and the RAFT models required an excess GPU memory for this experiment. The VMD-Net trained by [24] was on an RTX 3090 with 24GB of GPU memory; even after splitting the VMD-net and Raft model apart, reducing the batch size and using float point 16, the GPUs on the mlp teaching clusters and a GTX 3070 Ti. Thus, this model was trained on Google’s Cloud platform on a 40GB A-100 GPU. However, due to a limited budget, longer training regimes and hyperparameter tuning were hindered. Also, VMD is the smallest mirror detection model with 62M parameters, whereas PMD-Net had 147M and MirrorNet [24] had 121M.

In the labelled data creation part, initially, n (no of frames, mask pair to be added) was set to 10, but there was no change in the loss and evaluation metrics, and it was

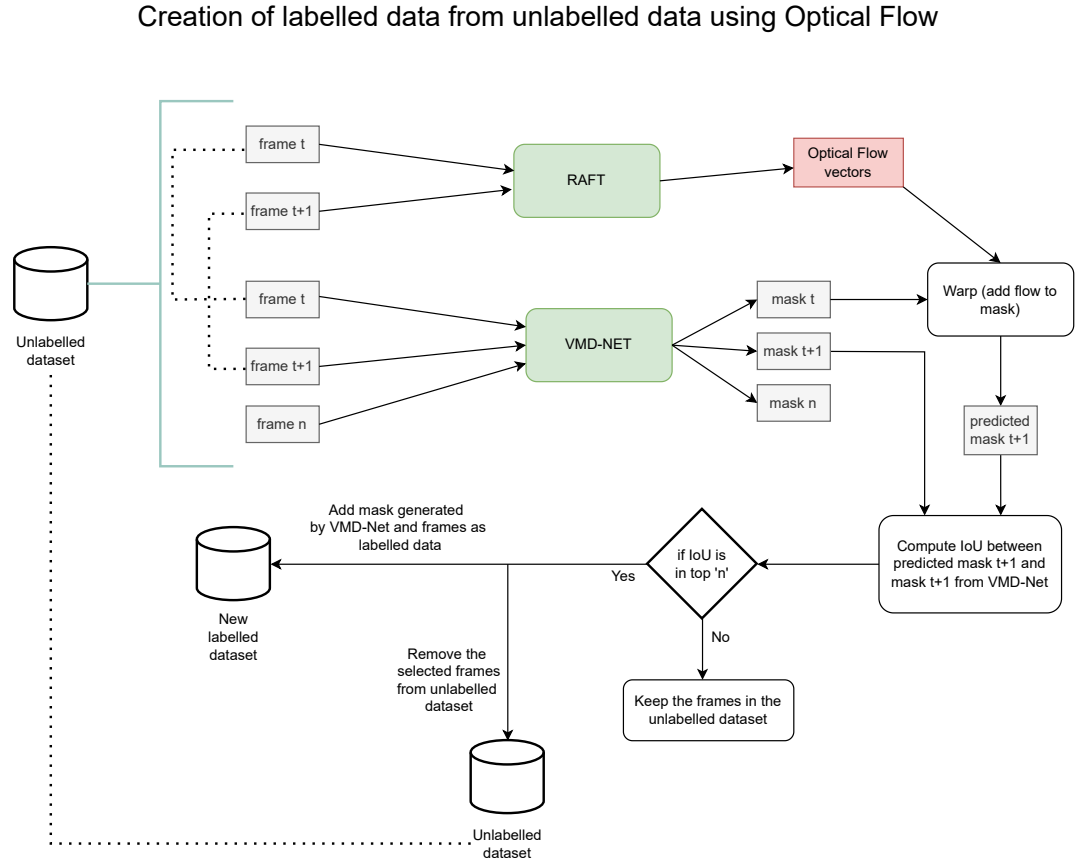


Figure 5.4: Creating new labelled data from unlabelled data. Using optical flow, mask at time t+1 is predicted and is compared with mask t+1 from VMD-Net. Highest N IoU data is added to the labelled dataset

later changed to 100.

In the updating part of the epoch, the default hyper-parameters were used, and only the starting learning rate was changed to 1e-4 from 1e-3 as, in finetuning, a lower learning rate is better as the model is almost at its convergence.

5.4.4 Results and Discussion

The IoU score improved slightly from 0.567 to 0.5681 when n was set to 100. MAE went from 0.1087 to 0.1084; pixel accuracy dropped from 0.8945 to 0.8934.

Note: The training was stopped unexpectedly, and only two epochs were completed before the compute units were exhausted. The model was trained for about 35 hours.

These are minimal changes in the metrics, and any proper conclusion cannot be

made if the designed training regime was effective or not.

We analysed the frame pairs selected by the above architecture, where Iou was the highest. About 30 frame pairs had good masks where the mirror was detected. About 60 per cent had mirrors, but other objects or small patches were also detected as mirrors. And the rest had some other objects. This means the labelled data created had false detection, and this learning method could degrade the model's performance. But another argument can be made that approximately 70 per cent of the data created had mirrors in it, so the model can learn to detect mirrors but with some false objects.

Chapter 6

Conclusion

This study focused on mirror detection in videos. In this thesis, we discussed why mirror detection is a difficult problem and showed how optical flow can be useful, especially for detecting mirrors. We also proposed a new large unlabelled dataset for video mirror detection, with over 4000 videos. Then, we did an extensive analysis of the optical flow on mirrors and found that using optical flow helps predict the mask movement, and the RAFT optical flow worked well on mirrors.

This study also experimented with the refinement of segmentation masks by a pretrained model, where we created and tested various optical flow features. We also found out that the pretrained VMD-Net model and PMD-Net model were overfitted in terms of IoU, where they had high IoU (over 0.9) for trained data and low IoU (around 0.56) for unseen data. The experiment hypothesis could not be concluded due to it.

Lastly, we propose a method to fine-tune pretrained mirror detection models using optical flow. In particular, we used the optical flow estimates to inspect if the predicted masks on unseen data are valid.

6.1 Limitations and Future work

The main limitation was the low performances of VMD-Net and PMD-Net models on unseen data; this hindered our experimentation. Also, these models were large and had a long training time. Hence, this research was computationally limited, even with modern graphics cards and mlp-clusters.

Like optical flow, depth can also be estimated by depth estimation models. This can again be a valuable feature for detecting mirrors.

The still scenes problem, where no optical flow is generated, can be addressed if a

set of consecutive optical flow images is used, like words in a n-gram. So, passing n optical flow features and n segmentation mask can address this issue but with the cost of larger model size and computation time. Further attention between the optical flow features and segmentation mask can also be implemented.

To solve the issue of the model not learning to use optical features, one can attempt to overlap the rough segmentation mask with the semantic optical flow features and use the segments with the highest IoU as a new mask. Similarly, zero-shot image segmentation models, like SAM, can be used to segment the original frames, where IoU between the rough segmentation mask and the segments could directly give the object itself.

To further refine the segmentation mask, morphological operations and contour smoothing can be used to fill out rough edges and holes. It was also observed that mirrors generally have distinct shapes, such as circles, ovals and rectangles. Even if only a part of the mirror is in the frame or an object is in front of it, remnants of the base shape are present. Adding a module which can leverage this in the early part of detection can also be explored.

Bibliography

[1] Coco dataset. https://cocodataset.org/home.

[2] Pexels a free stock image and video library. https://www.pexels.com.

[3] Pexels privacy policy. https://www.pexels.com/privacy-policy/.

[4] Pexels terms of service. https://www.pexels.com/terms-of-service/.

[5] Api documentation. https://www.pexels.com/api/documentation/, 2023. Accessed: May 2023.

[6] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks, 2018.

[7] Markus Braun, Sebastian Krebs, Fabian Flohr, and Dariu M Gavrila. Eurocity persons: A novel benchmark for person detection in traffic scenes. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1844–1861, 2019.

[8] Ron Brinkmann. *The Art and Science of Digital Compositing*. 2008.

[9] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. *European Conference on Computer Vision (ECCV)*, 2012.

[10] David Cochard. RAFT: A Machine Learning Model for Estimating Optical Flow, 2022. Accessed: 04-08-2023.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson Education, 2nd edition, 2002.

[17] Huankang Guan, Jiaying Lin, and Rynson WH Lau. Learning semantic associations for mirror detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5941–5950, 2022.

[18] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[19] Tianyu Huang, Bowen Dong, Jiaying Lin, Xiaohui Liu, Rynson WH Lau, and Wangmeng Zuo. Symmetry-aware transformer-based mirror detection. *arXiv preprint arXiv:2207.06332*, 2022.

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] V7 Labs. A guide to semantic segmentation, 2023.

[23] Fengze Li, Jieming Ma, Zhongbei Tian, Ji Ge, Hai-Ning Liang, Yungang Zhang, and Tianxi Wen. Mirror-yolo: An attention-based instance segmentation and detection model for mirrors. *arXiv preprint arXiv:2202.08498*, 2022.

[24] Jiaying Lin, Xin Tan, and Rynson WH Lau. Learning to detect mirrors from videos via dual correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9109–9118, 2023.

[25] Jiaying Lin, Guodong Wang, and Rynson WH Lau. Progressive mirror detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3697–3705, 2020.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[27] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pages 674–679. Morgan Kaufmann Publishers Inc., 1981.

[28] Haiyang Mei, Bo Dong, Wen Dong, Pieter Peers, Xin Yang, Qiang Zhang, and Xiaopeng Wei. Depth-aware mirror segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3044–3053, 2021.

[29] Eddie Montag. Visual and photographic optics - 13.2. geometric optics. https://www.cis.rit.edu/people/faculty/montag/vandplite/pages/chap$_1$3/ch13p2.$html$,$Nodate$.Acc
$September$7,2023.

[30] Nanonets. A deep dive into optical flow, 2019. Accessed: 2023-08-15.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[33] Jiaqi Tan, Weijie Lin, Angel X Chang, and Manolis Savva. Mirror3d: Depth refinement for mirror surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15990–15999, 2021.

[34] Xin Tan, Jiaying Lin, Ke Xu, Pan Chen, Lizhuang Ma, and Rynson WH Lau. Mirror detection with the visual chirality cue. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3492–3504, 2022.

[35] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020.

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[37] Xin Yang, Haiyang Mei, Ke Xu, Xiaopeng Wei, Baocai Yin, and Rynson WH Lau. Where is my mirror? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8809–8818, 2019.

[38] Oliver Zendel, Katrin Honauer, Markus Murschitz, Martin Humenberger, and Gustavo Fernandez Dominguez. Analyzing computer vision data-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1980–1990, 2017.

[39] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018.

Appendix A

VMD-Net on our dataset

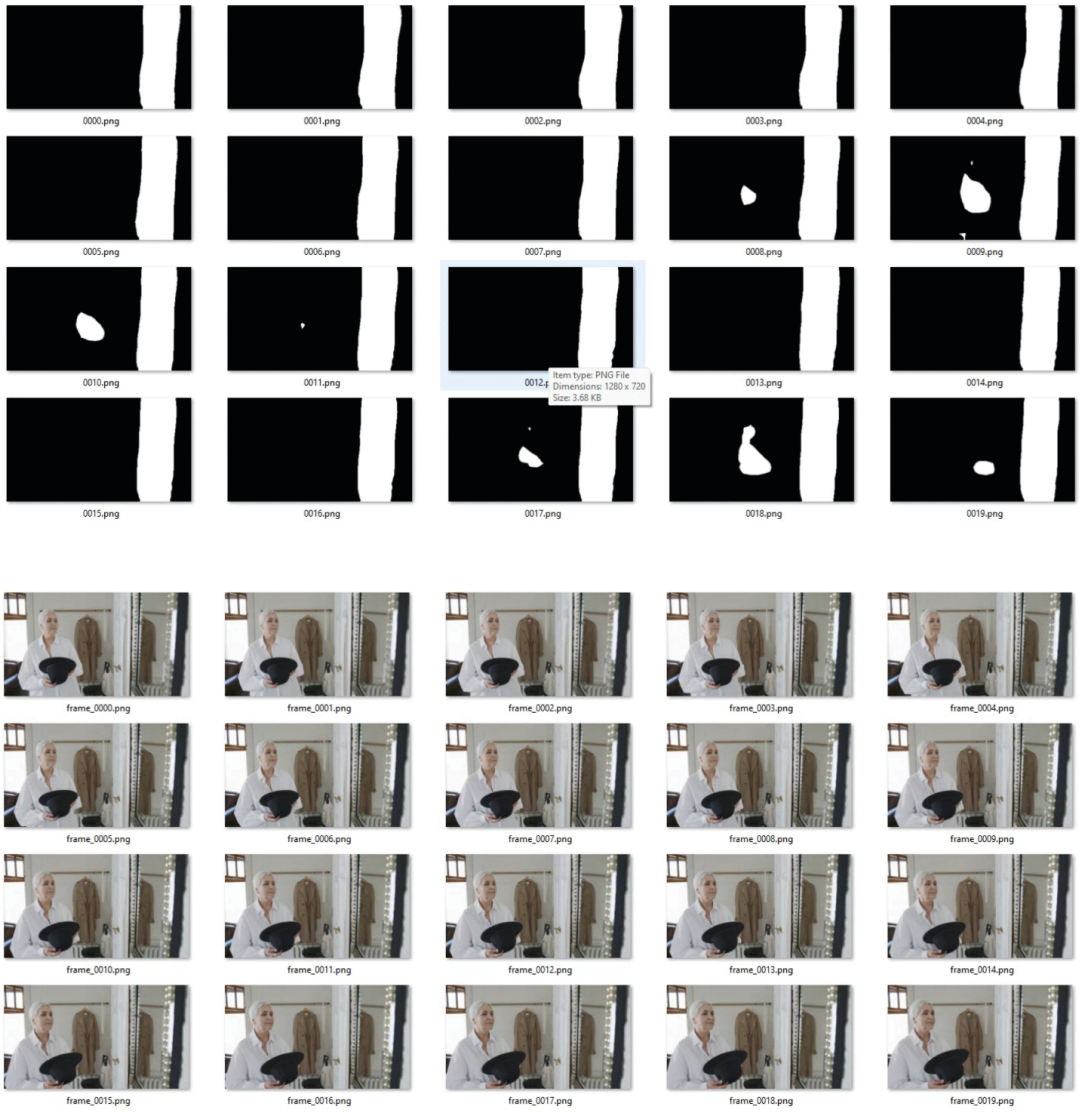


Figure A.1: Video frames (bottom) and binary mask made by VMD

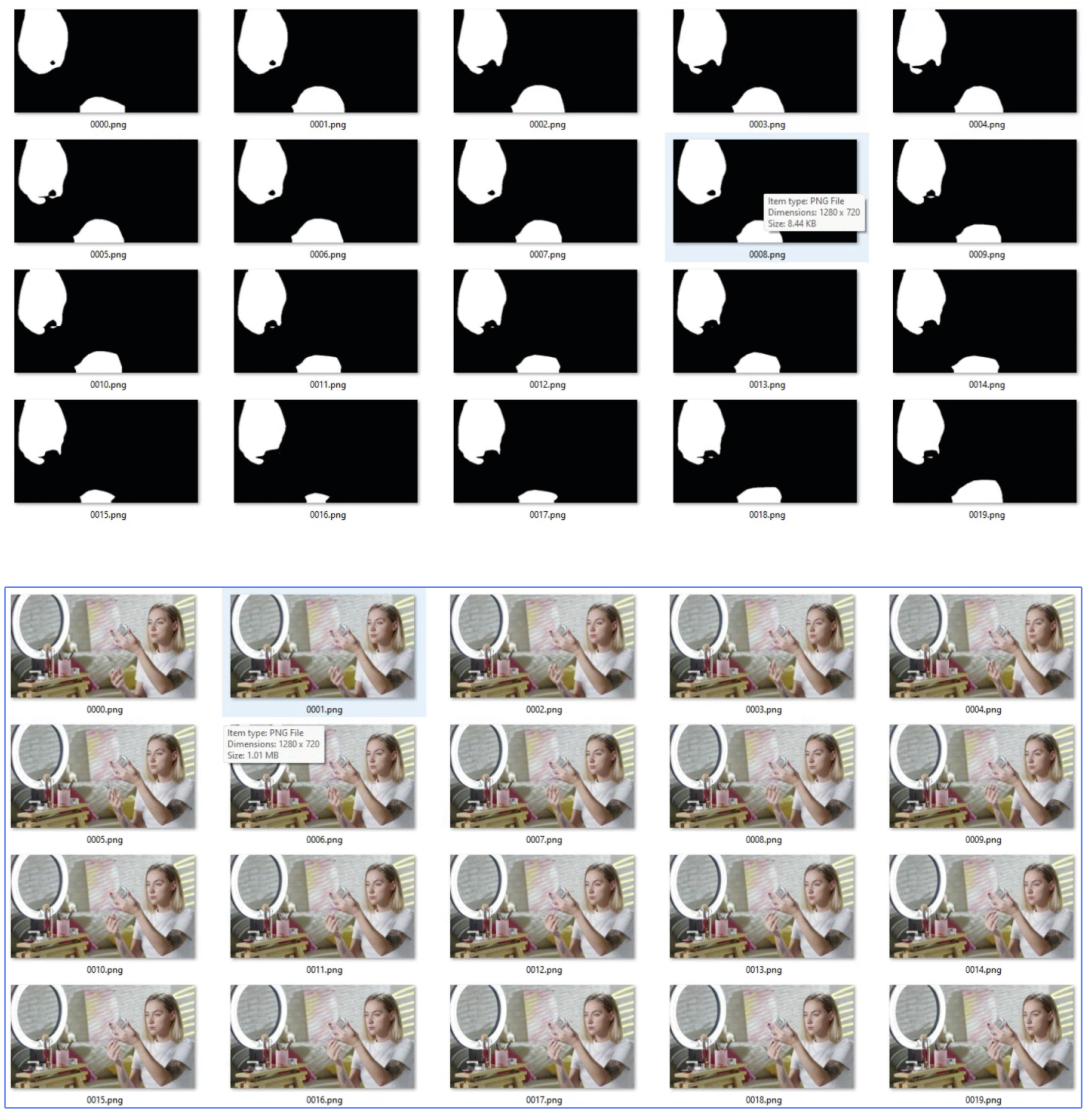


Figure A.2: Video frames (bottom) and binary mask made by VMD

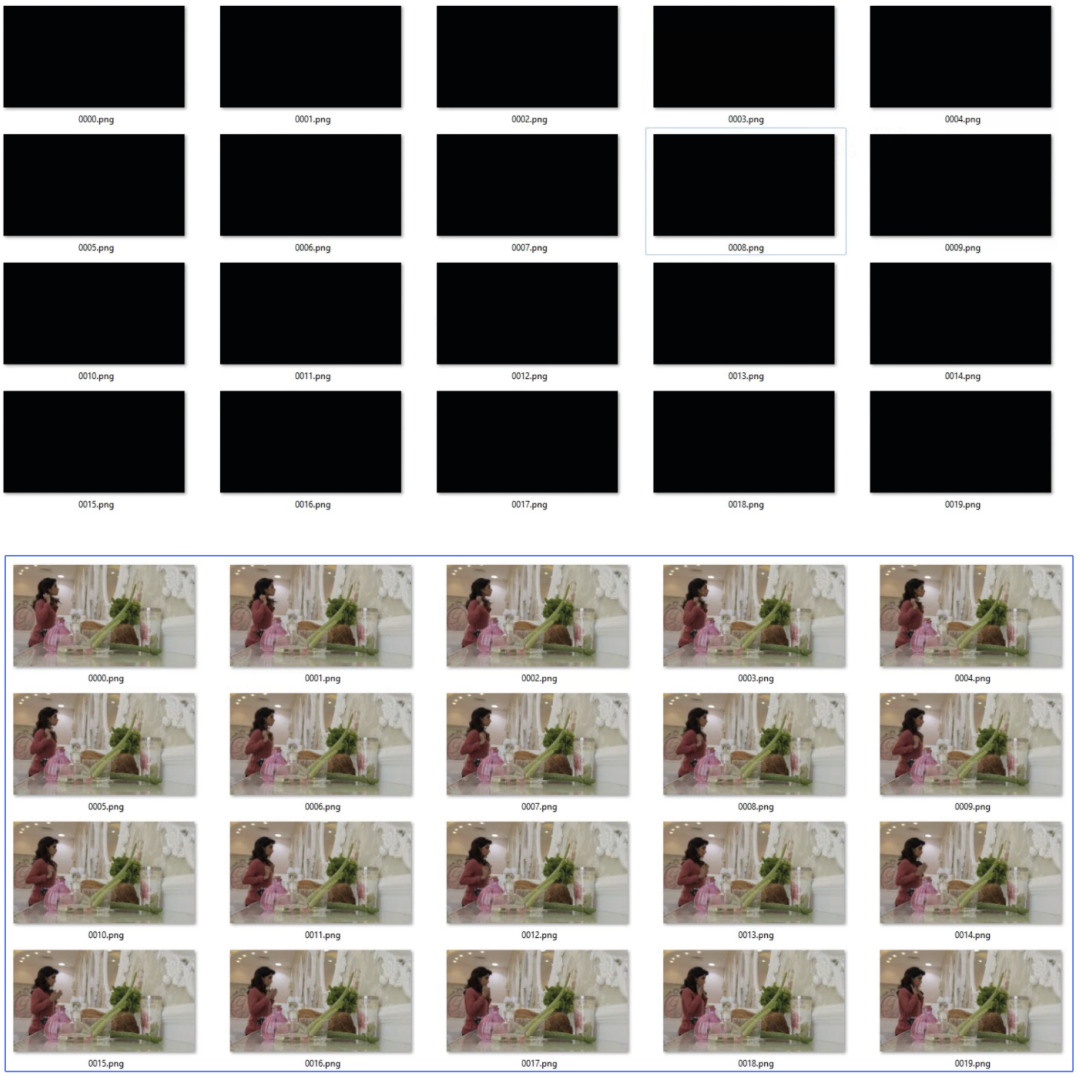


Figure A.3: Video frames (bottom) and binary mask made by VMD

Appendix B

Refinement model architecture

B.1 CAE

Encoder:

- Input: The forward function of the model takes two input images and concatenates them along the channel dimension. Hence, the input size is a 2-channel image (indicated by nn.Conv2d(2,...).
- Conv2d Layer: The first convolutional layer uses a 3x3 kernel and 32 filters. Padding is set to 1 to maintain the spatial size.
- ReLU Activation: Non-linearity added after the convolution.
- MaxPool2d: This layer halves the spatial dimensions (reduces height and width by a factor of 2).
- The same structure repeats two more times but with an increased number of channels: 32 to 64 and then 64 to 128.

Decoder:

- ConvTranspose2d Layer: The decoder uses transposed convolution (also known as deconvolution) to upsample the spatial size of the feature maps. The output padding=1 ensures that the spatial size doubles after each transposed convolution.
- ReLU Activation: Non-linearity added after the deconvolution.
- This structure repeats two more times. After every step, the depth of the feature maps is halved, and the spatial dimensions double.

- The final ConvTranspose2d layer reduces the depth to 1, which means the output is a single-channel image. The spatial size of the output will be the same as the input to the encoder due to the repeated upsampling.

B.2 U-Net

Encoder:

- 2 Convolutional layers with 64 filters, followed by a ReLU activation.
- MaxPooling for downsampling.
- 2 Convolutional layers with 128 filters, followed by a ReLU activation.
- MaxPooling for downsampling.
- 2 Convolutional layers with 256 filters, followed by a ReLU activation.
- MaxPooling for downsampling.
- 2 Convolutional layers with 512 filters, followed by a ReLU activation.
- MaxPooling for downsampling.

Decoder:

- Transposed Convolution to upsample to 256 channels.
- Convolutional layer with 256 filters, followed by a ReLU activation.
- Transposed Convolution to upsample to 128 channels.
- Convolutional layer with 128 filters, followed by a ReLU activation.
- Transposed Convolution to upsample to 64 channels.
- Convolutional layer with 64 filters, followed by a ReLU activation.
- Final Transposed Convolution to upsample to 1 channel and a Sigmoid activation function.