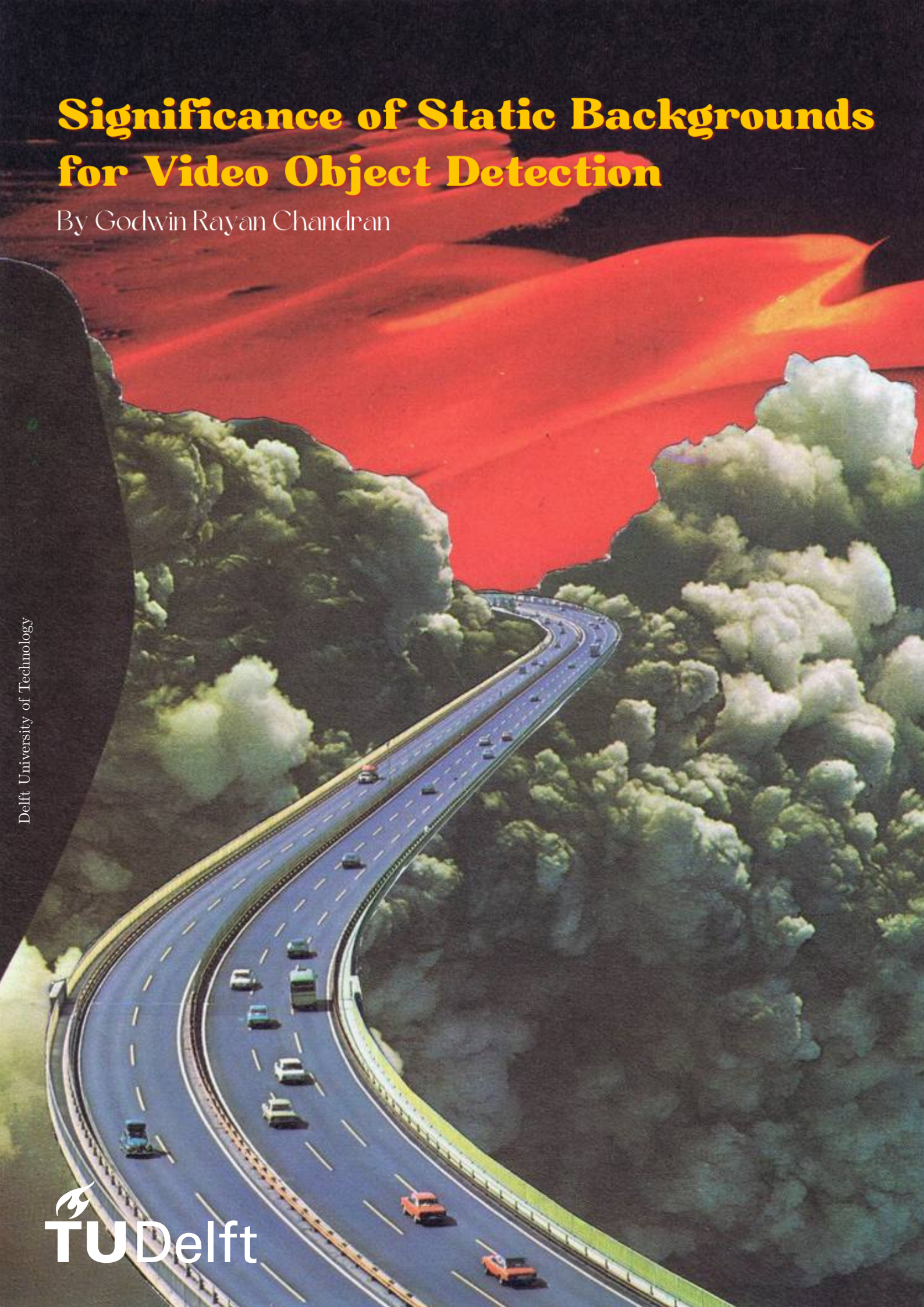
Significance of Static Backgrounds for Video Object Detection

By Godwin Rayan Chandran



Delft University of Technology

TU Delft

Significance of Static Backgrounds for Video Object Detection

by

Godwin Rayan Chandran

to obtain the degree of Master of Science in Robotics
at the Delft University of Technology,
to be defended publicly on Thursday October 13, 2022 at 9:00 AM.

Student number:	5347998	
Project duration:	November 1, 2021 – October 13, 2022	
Thesis committee:	Prof. dr. ir. Arkady Zgonnikov,	TU Delft, supervisor-3mE-CoR
	Prof. dr. ir. Jan van Gemert,	TU Delft, supervisor-EEMCS
	Prof. dr. ir. Holger Caesar,	TU Delft, 3mE-CoR

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This thesis marks my completion of the master's education in Robotics at the faculty of Mechanical, Maritime and Materials Engineering at TU Delft. The report documents the finding of my master thesis work.

Firstly, I would like to express my gratitude to Dr.Jan van Gemert for his motivation and support throughout the thesis. Your critical feedback helped me think and approach problems better. Secondly, I would to thank Ombretta Strafforello and Xin Liu. This thesis would have not been possible without their constant support. Words cannot explain how grateful I am to have them as my supervisors. I am also thankful to Dr.Arkady Zgonnikov for his helpful suggestions during the project. Finally, many thanks to Steve Nowee from AIIR Innovations for the feedback offered during the thesis.

I had the privilege to follow this master's education only because of the love and sacrifices of my family (Jesu Jasmine, Chandran Rayan and Godfrey Rayan). Thanks for being there at all times. Last but not least, thanks to Darshan, Stephy, Gopalan, Chinmay, Jeroen and all my friends in India and Delft.

Abstract

Video Object Detectors (VID) are used in various applications such as surveillance, inspection, etc. Often in these applications there exists a spatial area of interest and a static background. The static backgrounds remain constant throughout the video sequence in the training data establishing an undesirable correlation with the moving object during training. To hide static backgrounds in the video, masking is an option. We create multiple synthetic datasets and reveal that (i) VIDs detect moving objects better if the static background in the train and test set are similar or from the same distribution. (ii) VIDs drop in performance if the static background in the train and test are different. (iii) Adding more static backgrounds during training does not make VID robust to static background changes at test time. (iv) Masking or removing static backgrounds cannot prevent VIDs from learning correlations with static backgrounds. The experiments shed light on the usage of static backgrounds for detecting dynamic objects.

Contents

List of Figures

1

Scientific paper

Significance of Static Backgrounds for Video Object Detection

Godwin Rayan Chandran
Delft University of Technology
Delft, The Netherlands
g.r.chandran@student.tudelft.nl

Abstract

Video Object Detectors (VID) are used in various applications such as surveillance, inspection, etc. Often in these applications, a spatial area of interest and a static background exist. The static backgrounds remain constant throughout the video sequence in the training data, establishing an undesirable correlation with the moving object during training. To hide static backgrounds in the video, masking is an option. We create multiple synthetic datasets and reveal that (i) VIDs detect moving objects better if the static background in the train and test set are similar or from the same distribution. (ii) VIDs drop in performance if the static background in the train and test are different. (iii) Adding more static backgrounds during training does not make VID robust to static background changes at test time. (iv) Masking or removing static backgrounds cannot prevent VIDs from learning correlations with static backgrounds. The experiments shed light on the usage of static backgrounds for detecting dynamic objects.

1. Introduction

Generally, a video from a static camera comprises a region of interest accompanied by a static background. In the static background, a moving object does not appear. Object Detection (VID) models are trained on numerous videos with different static backgrounds. During training, VID models develop correlations between moving objects and static backgrounds. Consequently, the results of detecting moving objects differ when static backgrounds change at test time.

Assuming the VID model is to be implemented in different environments to detect moving objects, the model can encounter any static background. The previous works on image recognition relate the object with a background [4, 34, 42, 44]. For instance, a boat is more likely to appear on a blue background (water). Interchanging backgrounds between classes [44] or adding more backgrounds [36, 38] can be a solution for increased robustness. The problem arises when there exists no relation between an object and its static background. For instance, if we wish to detect boats in a warehouse, the blue background correlations are undesirable.

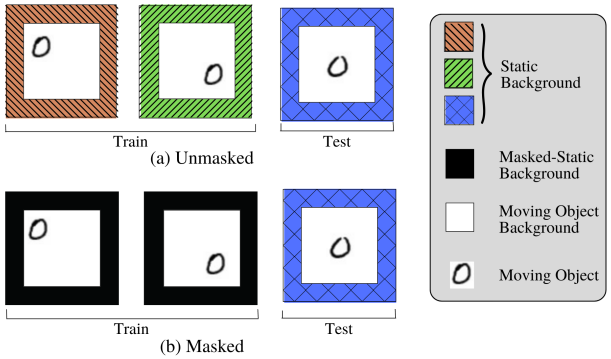


Figure 1. Proposed concept for the datasets. (a) Unmasked dataset with different static backgrounds, a white moving-object-background, and an MNIST digit [20] as a moving object. (b) Masked dataset with constant masked-static background, a white moving-object-background, and an MNIST digit as moving object. In both (a) and (b) the test set is unmasked for comparison purposes.

A common way to hide static background information in the dataset is masking [33]. A mask can be generated automatically [23, 25, 37] or manually designed based on the application's spatial area of interest. The mask is applied over each frame of the video to hide the static background. On the other hand, the static background information can be removed by cropping or camouflaging the static background with the moving-object-background. Here, moving-object-background refers to the region where the moving object appears or is probable to appear. It is unknown whether masking, cropping or camouflaging the static background during training or testing can make VIDs robust to static background changes.

In this paper, the aim is to expose the influence of static backgrounds on detecting moving objects. Since the current datasets [6, 8, 28] for VIDs are complex and non-

controllable for our setting (static camera video with different static backgrounds), we create fully-controlled synthetic datasets as shown in Fig.1. Each frame in the video dataset comprises three parts namely a moving object, a moving-object-background, and a static background. By altering the three parts individually, we gain insights into the extent of undesirable correlations in detecting moving objects under changing static backgrounds.

We make the following contributions. First, we design multiple datasets collectively named Static Background MNIST (SB-MNIST) to aid in understanding moving objects and static background relations better in VIDs. Second, We show that VID models detect moving objects better if the static background during training and test are similar or from the same distribution. Third, we reveal the undesirable correlation problem and its negative effect on the results when static backgrounds change at test time. Fourth, we demonstrate that masking or removing the static backgrounds does not make VIDs robust to static background changes at test time.

2. Related Work

We explore works related to the effects of backgrounds in VIDs. To the best of the author's knowledge, the effects of backgrounds in VIDs have not been investigated. Therefore, we focus on works related to the influence of backgrounds in image classification and image object detection. In literature, background refers to the combination of static background and moving-object-background (refer Figure.1). The term "static background" is also used in moving object detection [7] meaning a constant background.

2.1. Usage of backgrounds in object recognition

In the past decade, object detection algorithms [9, 17, 21] that exploit background information has demonstrated robust detection performances. The state-of-art object detection models utilize the background information efficiently, thereby surpassing each other on the benchmark datasets. The problem arises when the background differs from the training distribution. The set of correlations learnt during training is not beneficial anymore.

An interesting work on the most commonly used ImageNet [44], reveals backgrounds alone are sufficient to classify an object. Additionally, it provides an insight that changing background deteriorates the classification performances of state-of-the-art deep learning models. We follow similar assumptions in our experiment hypotheses changing backgrounds affect VID model performance.

2.2. Object Detection

Image Object Detection. Object detection with deep learning kick-started by splitting a video into images [13]. If an image is sent in its entirety for object detection it is called "one-shot" [2, 29–31, 35, 39–41] object detection method. On the other hand, if object proposals are extracted and fed to the network for object detection, it is called "two-shot" object detection method [12, 13, 32]. A drawback of image object detection is that it does not identify consistencies in objects between frames. As a result, they perform redundant computations and do not make use of temporal information.

Video Object Detection. VIDs were introduced to improve the accuracy of image object detectors by utilizing temporal information. VIDs use a group of frames for detecting objects. Unlike image object detectors categorising VIDs as one-shot and two-shot is not feasible as the approaches possess fewer similarities. Licheng et al [18] proposed a four-category classification based on the approach to capture temporal information. The four categories are based on Image Detection [14, 19], Motion information [16, 27, 45, 46], Effective Neural Networks [3, 24] and Features Filtering [43].

Firstly, Image Detection methods [14, 19] were introduced by adding a module to Image Object Detectors to capture temporal information. However, the ability to capture temporal information was limited as it is implemented only during test time. Secondly, the Motion information methods use additional models to capture motion information. Flow networks [11] possessed the ability to capture information. Inspired by flow network as an additional model, major works in VID Deep Feature Flow (DFF) [46] and Flow-Guided Feature Aggregation (FGFA) [45] were introduced. The main drawback of the flow-based methods was the increase in parameter size resulting in more training time. Thirdly, Efficient Neural Networks are methods that do not follow mainstream ideas and yet manage to lower redundant calculations and improve weaker features. Despite innovative approaches and complex architectures [24], the performance of Efficient Neural Networks is quite low compared to other methods.

Lastly, feature filtering methods temporally identify consistent features across video frames followed by aggregation of features to refine detection. One interesting benchmark using features filtering is Sequence Level Semantics Aggregation (SELSA) [43]. The working is similar to Faster-RCNN [32] but performs detection at the sequence level. SELSA utilizes region proposals from neighbouring frames termed reference frames. The semantic similarities are identified by generalized cosine similarity. After identifying the similarities, features are aggregated to obtain robust features. Since the method is based on the commonly used Convolutional Neural Networks, we employ it for our experiments. Additionally, the architecture of SELSA is simple and involves less training time.

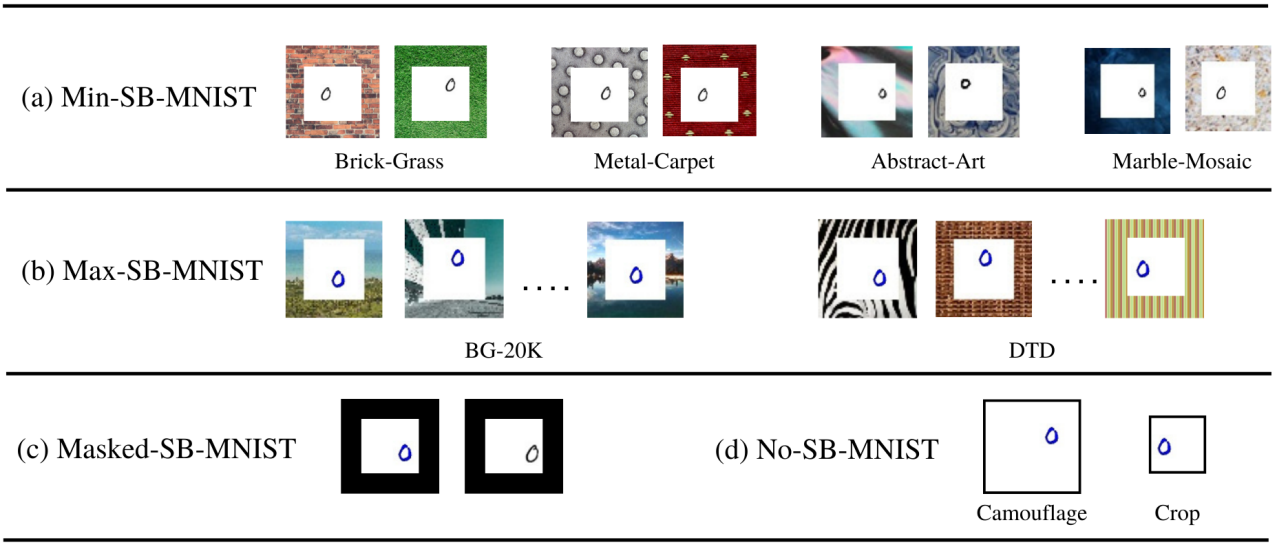


Figure 2. Overview of Static Background MNIST(SB-MNIST) (a) Four variants of Min-SB-MNIST in pairs namely Brick-Grass, Metal-Carpet, Abstract-Art and Marble-Mosaic. (b) Two variants of Max-SB-MNIST namely BG-20K and DTD. (c)Masked-SB-MNIST dataset with constant masked-static background and two variants are created with a black and blue moving object. (d) Two variants of No-SB-MNIST, Camouflage and Crop with no static background (Borders are added here for visualization purposes).

3. Methodology

To identify the influence of static backgrounds in VIDs, the first step is to choose a suitable dataset for our experiments. A majority of video sequences in large-scale datasets like ImageNet VID [8], EPIC KITCHENS [6] and YouTube BoundingBoxes [28] are recorded with moving cameras. This makes the datasets unsuitable for our setting. Initially, we identify a dataset introduced for moving object recognition [26]. The videos are recorded using static cameras in real-life scenarios. However, the dataset has a few disadvantages. First, the moving object classes are imbalanced (13,442 cars and 25,385 person). Second, the occurrences of moving objects and static backgrounds are not balanced. So we decide to build synthetic datasets for the experiments.

We carefully build different balanced datasets for the experiments. Each dataset has B static backgrounds, K moving object classes, N moving object instances and a constant moving-object-background. For the moving objects, we employ the MNIST [20] digits dataset. The MNIST dataset has handwritten digits from 0 to 9 (Therefore, K=10). The datasets are collectively named Static Background-MNIST (SB-MNIST).

The moving object's initial location is randomized inside the moving-object-background. Each moving object class (K_i) is associated with a distinct motion pattern as shown in Figure.3. By random initialization and distinct motion assignment, we aim to prevent bias in start location and motion. The object moves at a rate of one pixel per frame in the video. If the object exceeds the boundaries of the moving-object background, the digit is reset randomly inside the moving-object-background.

The SB-MNIST dataset has a total of 1000 videos, where 600 videos belong to the training set, 200 videos to the validation set, and 200 videos to the test set. All the videos are of equal length of 50 frames. The dimension of each frame is 64x64. A 40x40 white moving-object-region is located at the centre of the frame and surrounded by static background. To note, in this dataset we assume that the static background remains constant for the entire sequence of a video.

Min-SB-MNIST. As the name indicates Minimum-SB-MNIST datasets possess the least number of unmasked static backgrounds. The datasets are composed of 2 static backgrounds for the train, validation and test sets. We create four variants in the Min-SB-MNIST format namely Brick-Grass, Metal-Carpet, Abstract-Art and Marble-Mosaic.

Max-SB-MNIST. To simulate a realistic scenario where the unmasked static backgrounds are different for each video, we introduce Maximum-SB-MNIST. In Max-SB-MNIST, the number of static backgrounds B is equal to the number of videos in the train, validation and test set. We create two variants based on the dataset used for the static backgrounds. In the first variant, We pick 600 static backgrounds for training and 200 static backgrounds for testing from the BG-20K [22] test set. The validation set

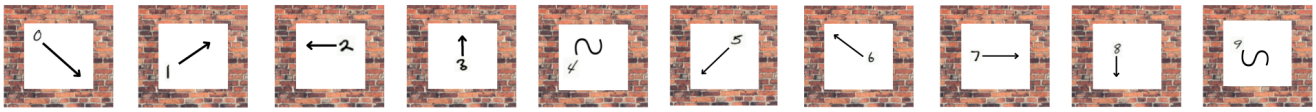


Figure 3. Motion associated with each MNIST digit class(0 to 9) in SB-MNIST dataset. The arrow marks indicate the direction of movement. If the moving object (MNIST digit) exceeds the moving-object-background (white region), it is reset randomly inside the moving-object-background.

static backgrounds are a subset of the training background i.e 200 backgrounds from training are used. For the second variant, 600 static backgrounds for training and 200 static backgrounds for testing are chosen from 10 texture categories of the Describable Texture Dataset(DTD) [5].

Masked-SB-MNIST. To hide the static backgrounds, we mask the dataset i.e convert the pixel values of the static regions to 0. By masking, we attain a constant static background for train, validation, and test set. We create two variants of Masked-SB-MNIST. One with a black moving object and the other a blue moving object.

No-SB-MNIST. We replace the static background with the colour of the moving-object-background. After replacement, there is no static background in the dataset. We name this variant of No-SB-MNIST as camouflage. Similarly, we create another variant called crop, where we remove the static background by cropping. Therefore, the dimensions of the crop variant are equal to the dimension of moving-object-background i.e 40x40.

The statistics of each dataset are listed in Table.1 and example frames from each dataset are shown in Figure.2.

Dataset	Type	B	N
Min-SB-MNIST	Train	2	60
	Validation	2	20
	Test	2	20
Max-SB-MNIST	Train	600	60
	Validation	200	20
	Test	200	20
Masked-SB-MNIST	Train	1	60
	Validation	1	20
	Test	1	20
No-SB-MNIST	Test	0	20

Table 1. Statistics of SB-MNIST. The type refers to the purpose of the dataset by the VID model i.e train, validation or test. B is the number of static backgrounds. N is the number of instances. N/K gives the number of instances per class.

4. Experiments

Implementation details for VID model. For experiments, we use one of the state-of-the-art VID methods, SELSA [43]. The SELSA model is composed of a Feature Network, Detection Network, and a SELSA module. First, in the Feature Network, we use ResNet-50 [15] pre-trained on the ImageNet [8] dataset. For training, the first layer is frozen to reduce over-fitting. Second, in the Detection Network, the anchor scales and aspect ratios of the Region Proposal Network [32] are modified to accommodate the small MNIST digits. Stochastic Gradient Descent training is implemented with a batch size of 1 on TeslaV100 GPU. For evaluation metrics, we use mean Average Precision(mAP) [10].

4.1. Performance on same static backgrounds

Same pair of static backgrounds. The motivation behind the experiment is to record the performance of VID with no static background variations. During model deployment, the static backgrounds that were present during training are expected to be present. We simulate the scenario using the same pairs of static backgrounds at train and test time. The same pair scenario is considered the most favourable situation and peak detection results are expected.

We employ four datasets of Min-SB-MNIST. The VID is individually trained on each dataset namely Brick-Grass, Metal-Carpet, Abstract-Art and Marble-Mosaic. The VID model is optimized with a validation set with the same pairs of static backgrounds. Finally, the four trained VID models are tested with their corresponding test sets. The results are shown in Table.2 along the principal diagonal in the first four rows (highlighted in yellow). We obtain an mAP value greater than 60 in all the cases. We do not achieve an mAP close to 100 due to the lower number of instances and random initialization of moving objects.

Same distribution of static backgrounds. Video datasets are often representative of the real world [6, 8, 28]. Each video is recorded with static backgrounds belonging to different indoor and outdoor environments. The aim of the experiment is to record the performance when the VID model is expected to have static backgrounds at test time that are similar to the training distribution. We establish the same distribution scenario by using static backgrounds be-

Trained on	Tested on mAP% (IoU=0.50:0.95)									
	Metal-Carpet	Brick-Grass	Abstract-Art	Marble-Mosaic	BG-20K	DTD	Masked-SB	Camou-flage	Crop	**Mean drop%**
Metal-Carpet	**61.4**	58.9	52.2	56.3	46.2	44.3	50.0	47.6	15.4	24.5
Brick-Grass	31.3	**61.6**	19.0	40.7	49.2	27.4	26.5	51.2	18.7	46.4
Abstract-Art	39.2	57.1	**63.6**	53.6	43.7	41.5	50.9	43.6	16.1	32.0
Marble-Mosaic	30.3	60.9	50.2	**64.6**	47.8	45.2	56.5	46.7	16.3	31.5
BG-20K	38.8	58.1	50.5	56.8	**57.1**	44.1	40.1	27.2	19.8	26.6
Masked-SB	27.2	56.5	33.2	49.3	27.0	20.0	**64.6**	36.0	29.3	46.1

Table 2. VID performance(mAP) for detecting moving object classes 0 to 9 when training on one dataset variant (rows) and testing on another (columns). Metal-Carpet, Brick-Grass, Abstract-Art and Marble-Mosaic are variants of Min-SB-MNIST. BG-20K and DTD are variants of Max-SB-MNIST. Camouflage and Crop are variants of No-SB-MNIST. Mean drop% is calculated with results other than training and testing with the same variant.

longing to similar indoor and outdoor environments. The same distribution is the favourable situation for per video static background scenarios.

To execute this experiment, We train and test with the BG-20K variant of Max-SB-MNIST datasets. We obtain an mAP of 57.1 (highlighted in green in Table.2) which is lower when compared to training and testing on the same pair of backgrounds. The lower mAP is due to changing static backgrounds at test time which are similar but not the same.

4.2. Effects of different static backgrounds

Different pairs of static backgrounds. Can VID models perform well when exposed to a static background pair that is not in the training set? In datasets with a limited variety of static backgrounds, the VID models are more likely to over-fit the static background. The main aim of the experiment is to identify if static backgrounds are exploited during train time and how severe VID models over-fit the static background. We observe the effects by testing the pre-trained models from Experiment.4.1 on different pairs of static backgrounds.

To begin with, we utilize the VID models trained individually on four dataset variants of Min-SB-MNIST. We test the VID on every other variant that possesses different static backgrounds. For instance, if the VID model is trained on the brick-grass variant then the model is tested on other variants Metal-Carpet, Abstract-Art and Marble-Mosaic. We compare the results with the same static background results obtained from Experiment.4.1. The results (highlighted in red in Table.2) reveal that changing static backgrounds during testing reduces the performance in all cases. The performance drop affirms that undesirable correlations are established between static background and moving objects during training. The performance drop occurs to an extent of 69%. The performance decrease explains the extent of exploitation of static backgrounds detecting moving objects. Among the trained VID models, the model trained on Metal-Carpet works best when compared to others. The reason could be that the Metal-Carpet variant possesses static background features similar to other variants. Sample detection results are shown in Figure.4.

Different distribution of static backgrounds. Consider a scenario where a VID model is trained on a dataset possessing different static backgrounds in each video. If the trained VID model is implemented in different applications, VID models will encounter different environments for each application at test time. So the question is: Can VID models perform well when exposed to a different static background distribution?

Initially, we employ the VID model trained on the BG-20K variant of Max-SB-MNIST. We test BG-20K trained VID model with the DTD variant of Max-SB-MNIST. BG-20K comprises indoor and outdoor scenes whereas DTD has texture static backgrounds. When comparing with the same distribution results, the performance decreases (highlighted in blue in Table.2) if the static backgrounds are from a different distribution. Thus, VIDs cannot perform well when the static backgrounds are different or from a different distribution.

4.3. More static backgrounds Vs Less static backgrounds

The motivation for this experiment is to identify if using more static backgrounds is an advantage for VID models. To describe in another context, we check if data augmentation is a solution to the static background exploitation problem.

Train with more, test with less. The Max-SB-MNIST with 600 static backgrounds during training offers a form of data augmentation accompanied by regularization. Ultimately, the VID model trained on Max-SB-MNIST is less likely to over-fit the static backgrounds. Considering the advantage of augmentation, we record the results (high-

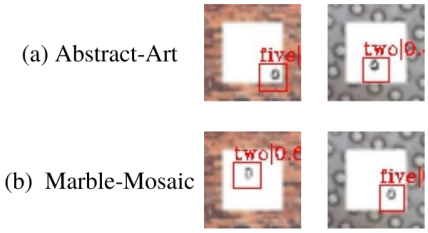


Figure 4. Bounding box detection results of VID model. (a) Results of VID model trained on Abstract-Art variant and tested on brick-grass and metal-carpet (b) Results of VID model trained on Marble-Mosaic variant and tested on Brick-grass and metal-carpet. In both (a) and (b) the detections are False Positives.

lighted in purple) of training with more static backgrounds (BG-20K) and testing with less static backgrounds (Min-SB-MNIST). We compare the mAP with results of Experiment.4.2 (highlighted in red) in Table.2. It is observed that Min-SB-MNIST trained models surpass the Max-SB-MNIST trained model in the majority of the cases except for the Marble-Mosaic variant. Therefore, adding more static backgrounds during training does not necessarily improve a VID model's robustness to static backgrounds at test time.

Train with less, test with more. Consider a situation where we train a VID model with 2 static backgrounds. Will the trained VID model perform well when there are 200 different static backgrounds at test time? To answer this question, we employ the Min-SB-MNIST trained VID models. At test time, we expose the VID models to 200 static backgrounds from the DTD variant of Max-SB-MNIST. The results are in Table.2 (highlighted in pink) demonstrate that in two variants, Metal-Carpet and Marble-Mosaic work better than Max-SB-MNIST (highlighted in blue) trained with 600 backgrounds. From this comparison, we conclude that using less static backgrounds can perform better occasionally. We mention occasionally since Min-SB-MNIST trained (Brick-grass and Abstract-Art) performs worse than Max-SB-MNIST.

4.4. Can masking reduce the influence of static backgrounds?

In section.4.1, 4.2 and 4.3, we investigated the effects of unmasked static backgrounds. In this experiment, we mask the static backgrounds. By masking, the static backgrounds in all videos are converted to one constant zero-valued black static background. The assumption for masking is that homogeneous background reduces the complexity of the static background. We define a masked static background as less complex as there is no change in pixel values in the static background.

First, we train with the VID model with the Masked-SB-MNIST dataset. The Masked-SB-MNIST trained VID model is tested on four unmasked variants of Min-SB-MNIST. The results show that masking the static backgrounds during training does not make VID models robust to background changes. The mAP scores are reduced to an extent of more than 50% compared to testing on the same static backgrounds. Similar poor results were observed when testing Masked-SB-MNIST with Max-SB-MNIST variants BG-20K and DTD with static backgrounds per video. Masking during training does not improve the detection of the moving object since VIDs overfit the masked static background relatively more than Max-SB-MNIST. Additionally, the masked static background being a simple homogeneous static background does not equip VID models to face complex heterogeneous static backgrounds at test time.

Second, we implement masking at test time. The VID models trained on unmasked Min-SB-MNIST are tested on Masked-SB-MNIST. The results drop to an extent of more than 50%. Although VID models are trained on complex backgrounds their performance decreases even if a less complex constant static background is encountered at test time. Therefore, masking static backgrounds during training or testing does not reduce background influence in VIDs.

4.5. Do VIDs perform better if the static backgrounds are removed?

Camouflage static background with moving-object-background. Since VIDs cannot generalize well to different static backgrounds at test time, can removing static backgrounds improve detection results? In other words, does harmonizing the static background with the moving-object-region improve the performance of VIDs? The assumption is that camouflaging static backgrounds with moving-object-background will reduce the visual distraction in the scene [1].

The models trained on four Min-SB-MNIST variants, Max-SB-MNIST(BG-20K) and Masked-SB-MNIST are tested on Camouflage variant No-SB-MNIST. Results drop to a maximum of 31% for Min-SB-MNIST, 54% for Max-SB-MNIST and 44% for Masked-SB-MNIST. From the results, we observe that the absence of static background affects performance similar to the presence of a different background at test time. Therefore, reducing distractions from static backgrounds during test time cannot improve the detection of moving objects.

Crop static background. In this experiment, we remove the static background by cropping at test time. We utilize the models trained on Min-SB-MNIST, Max-SB-MNIST and Masked-SB-MNIST. We test the models on the Crop variant of No-SB-MNIST. When compared to the other variants, the test mAP scores of the Crop variant are the lowest. The test results drop on average by 69.1%.

A possible reason for poor results could be due to the exploitation of the absolute spatial location of the static background. Cropping during testing disrupts the location information learnt in training i.e a static background always occurs around a moving-object-background.

5. Limitations and Conclusion

In this paper, we reveal the static background influence in VID using the SB-MNIST dataset. The SB-MNIST is modular and extendable. Our dataset has no limitations in terms of simulating different settings. However, it is unknown how well the synthetic datasets translate to real-life datasets.

To simplify the experiments, we considered fixed boundaries for static backgrounds and moving-object-background. Additionally, we apply constant colour for the moving-object-background. Future experiments using SB-MNIST would be to identify the behaviour of VID models to static backgrounds under different dataset sizes, static-background boundaries, moving-object-background and moving object colours. We consider only the minimum of 2 and the maximum number of 600 static backgrounds. Analyzing the effects of the number of static backgrounds for VIDs is left for future work.

VID models are rapidly evolving and registering their competence on large-scale datasets. However, they are not equipped to face undesirable correlation problems in real-life applications. To solve the problem, it is not feasible to add all possible static backgrounds to the training dataset to make robust VIDs. Therefore, an opportunity to solve the undesirable correlation problem would be along the VID model itself. Integrating VIDs with methods related to image inpainting or efficient video processing that treat static backgrounds as invalid or redundant information could be a possible direction for solving the problem. If consistent results are achieved on SB-MNIST, the VID model is robust.

To conclude, the intention of SB-MNIST and the experiments was to disclose the problems associated with static backgrounds. First, we demonstrate that VID performs better if deployed with static backgrounds same or similar to the training datasets. Second, we show that if there is a change in the static background at test time, the results drop drastically. A shortcut to solve the performance drop is to replace the static backgrounds at test time with static backgrounds from the training set. Third, we identify that training with lesser static backgrounds can produce results occasionally better than a model trained with more static backgrounds. Finally, we mask and remove the static backgrounds in an attempt to reduce the influence of static backgrounds in VIDs. Yet, we face a drop in performance, concluding masking or removing static backgrounds cannot prevent VIDs from exploiting static backgrounds.

References

[1] Kfir Aberman, Junfeng He, Yossi Gandelsman, Inbar Mosseri, David E. Jacobs, Kai Kohlhoff, Yael Pritch, and Michael Rubinstein. Deep saliency prior for reducing visual distraction. *CoRR*, abs/2109.01980, 2021. 6

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. 4 2020. 2

[3] Ting-Wu Chin, Ruizhou Ding, and Diana Marculescu. Adascale: Towards real-time video object detection using adaptive scaling. 2 2019. 2

[4] Myung Choi, Antonio Torralba, and Alan Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33:853–862, 05 2012. 1

[5] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 4

[6] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 3, 4

[7] Angel P. del Pobil and Ester Martínez. *Robust Motion Detection in Real-Life Scenarios*. 01 2012. 2

[8] 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. 1, 3, 4

[9] Carl Doersch, Abhinav Gupta, and Alexei Efros. Context as supervisory signal: Discovering objects with predictable context. pages 362–377, 09 2014. 2

[10] Mark Everingham, · Luc, Van Gool, · Christopher, K I Williams, John Winn, Andrew Zisserman, M Everingham, L Van Gool, K U Leuven, Belgium C K I Williams, J Winn, and A Zisserman. The pascal visual object classes (voc) challenge. 4

[11] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. 4 2015. 2

[12] Ross Girshick. Fast r-cnn. 4 2015. 2

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 11 2013. 2

[14] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S. Huang. Seq-nms for video object detection. 2 2016. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 4

[16] Yongyi Lu HKUST, Cewu Lu, and Chi-Keung Tang HKUST. Online video object detection using association lstm. 2

[17] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. pages 3588–3597, 06 2018. 2

[18] Licheng Jiao, Ruohan Zhang, Fang Liu, Shuyuan Yang, Biao Hou, Lingling Li, and Xu Tang. New generation deep learning for video object detection: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 2

[19] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. 4 2016. 2

[20] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 1, 3

[21] Jiaxu Leng, Yihui Ren, Wen Jiang, Xiaoding Sun, and Ye Wang. Realize your surroundings: Exploiting context information for small object detection. *Neurocomputing*, 433, 01 2021. 2

[22] Jizhizi Li, Jing Zhang, Stephen J Maybank, and Dacheng Tao. Bridging composite and real: Towards end-to-end deep image matting. *International Journal of Computer Vision*, 2022. 3

[23] Long Ang Lim and Hacer Keles. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. *Pattern Recognition Letters*, 112, 01 2018. 1

[24] Mason Liu, Menglong Zhu, Marie White, Yinxiao Li, and Dmitry Kalenichenko. Looking fast and slow: Memory-guided mobile video object detection. 3 2019. 2

[25] Murari Mandal, Vansh Dhar, Abhishek Mishra, and Santosh Vipparthi. 3dfr: A swift 3d feature reductionist framework for scene independent change detection, 12 2019. 1

[26] Murari Mandal, Lav Kush Kumar, Mahipal Saran, and Santosh Vipparthi. Motionrec: A unified deep framework for moving object recognition. 03 2020. 3

[27] Huizi Mao, Taeyoung Kong, and William J. Dally. Catdet: Cascaded tracked detector for efficient object detection from video. 9 2018. 2

[28] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. *CoRR*, abs/1702.00824, 2017. 1, 3, 4

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. 6 2015. 2

[30] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. 12 2016. 2

[31] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. 4 2018. 2

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 6 2015. 2, 4

[33] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23:309–314, 08 2004. 1

[34] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk — quantifying and controlling the effects of context in classification and segmentation. pages 8210–8218, 06 2019. 1

[35] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. 11 2019. 2

[36] Koya Tango, Takehiko Ohkawa, Ryosuke Furuta, and Yoichi Sato. Background mixup data augmentation for hand and object-in-contact detection, 02 2022. 1

[37] M. Tezcan, Prakash Ishwar, and Janusz Konrad. Bsuv-net: A fully-convolutional neural network for background subtraction of unseen videos. pages 2763–2772, 03 2020. 1

[38] Maoqing Tian, Shuai Yi, Hongsheng Li, Shihua Li, Xuesen Zhang, Jianping Shi, Junjie Yan, and Xiaogang Wang. Eliminating background-bias for robust person re-identification. pages 5794–5803, 06 2018. 1

[39] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. 11 2020. 2

[40] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh. Cspnet: A new backbone that can enhance learning capability of cnn. 11 2019. 2

[41] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. 5 2021. 2

[42] Zilei Wang, Dao Xiang, Saihui Hou, and Feng Wu. Background driven salient object detection. *IEEE Transactions on Multimedia*, PP:1–1, 12 2016. 1

[43] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Sequence level semantics aggregation for video object detection. 7 2019. 2, 4

[44] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition, 06 2020. 1, 2

[45] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. 2

[46] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. 11 2016. 2

2

Introduction

The fundamental tasks of a robot is to sense the environment, process sensory information, and act accordingly. This process of sensing and processing is anonymous with visual recognition in computer vision. Visual recognition has several categories among which classification, object detection and semantic segmentation using Convolutional Neural Networks(CNNs) are gaining traction in various applications. Object detection involves identifying what(classification) and where(localization) is the object in a given image or video. For years, object detection is carried out frame by frame. The video is split into images and object detection is performed on each image. Image object detectors have proven to work quite well through the years. More recently Video Object Detectors(VIDs) have evolved with detection performed utilizing a stack of images.

For instance, consider an industrial inspection system where a static camera captures an image or video in a moving conveyor belt as shown in 2.1. The camera data is fed as an input to the object detection model that transforms the data in a way to extract prominent features. The objects in the scene are displayed in the form of bounding boxes and confidence scores. Ultimately, the inspection system is aware of the environment and can perform necessary communication.

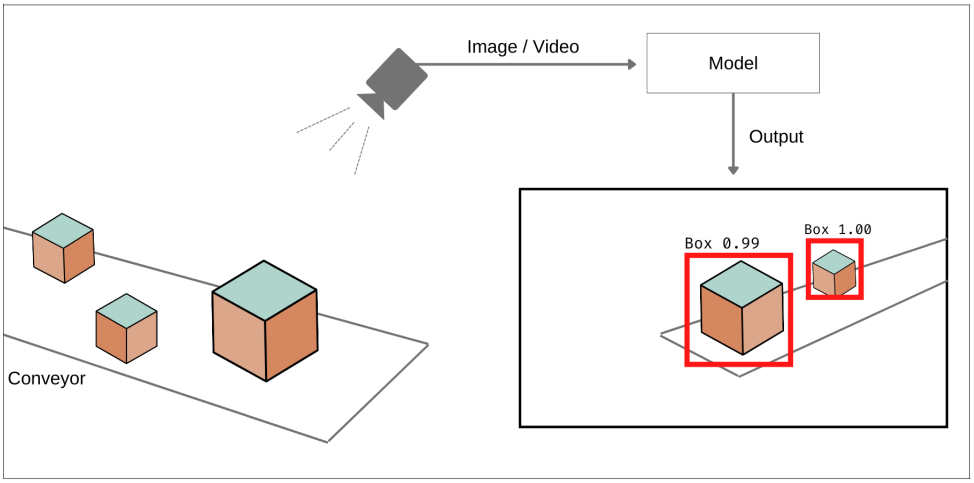


Figure 2.1: Illustration of an industrial inspection system. A static camera captures the boxes on the conveyor and feeds the input as an image/video to the model. The model extracts relevant features and outputs results in the form of a bounding box with a confidence score.

2.1. Problem

In many systems with a static monocular camera, the spatial area of interest depends on the moving object. As the video is larger than the spatial area of interest there exists static background throughout its sequence. In the static background, the moving object does not appear. Object detection models are trained on numerous videos with different static backgrounds. During training, correlations are

developed between the moving object and its static background. Consequently, the detection results on objects deviate when tested with static backgrounds out of the training distribution. The correlation is desirable in certain situations when the moving object class is associated with a static background. For instance, a fish is more likely to appear on a blue static background. The problem arises when there is no relation between the moving object and the static background. For instance, consider training a model to detect fish in an industry conveyor. The blue background correlation is undesirable as the industry static background is different.

The above-mentioned problem can be solved by collecting video data of moving objects with diverse static backgrounds. But in reality, it is not possible to obtain all possible static backgrounds. One of the common methods to hide static background information is masking. Masking is a process of converting image pixel values to zero. These masks can be manually drawn or automatically generated over irrelevant regions. Yet, it is not known how video object detectors use static backgrounds for moving object detection and will masking it benefit detection performance.

For this thesis, we get observations from problems in borescope inspection at AIIR Innovations. The borescope inspection system has a static camera to inspect damages in a rotating blade. We break down a video frame into three parts namely a static background, moving-object-background and moving object(refer Fig.2.2). From Fig.2.2, we observe that the static background at time *t*, *t+1* and *t+2* remains constant whereas the moving object and moving-object-background keep changing in every frame.

Consider a situation where there are borescope inspection videos with different static backgrounds at train time(shown in yellow and red in Fig.2.3). During training, the VID model develops a correlation with the static backgrounds. The results deviate when the model is exposed to a different static background at test time(shown in purple in Fig.2.3. Therefore, the correlations with the static backgrounds are undesirable.

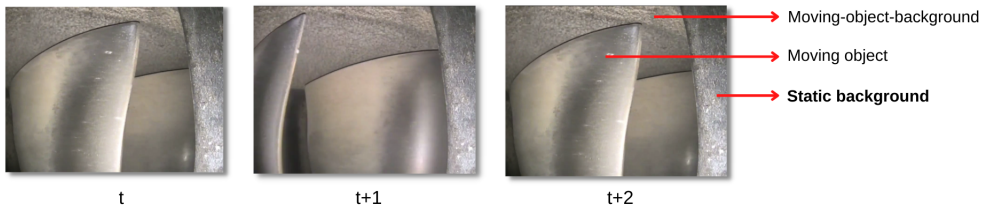


Figure 2.2: Video frames from borescope inspection at different time intervals. We split the video into three parts namely a static background, moving-object-background and moving object.
[1]

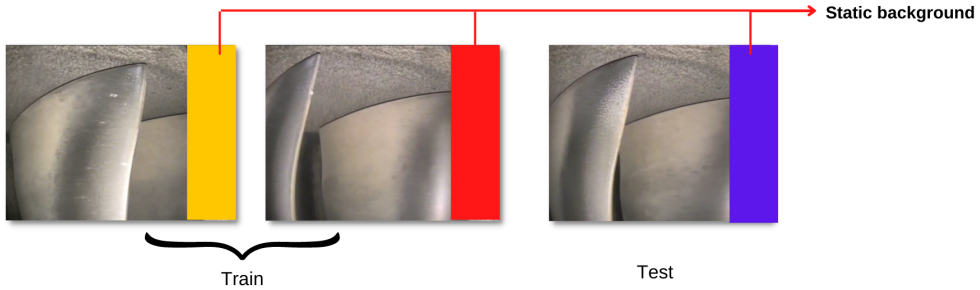


Figure 2.3: Illustration of different backgrounds at train and test time(for visual purposes only). The static backgrounds at train time are shown in yellow and red. The static background at test time is shown in purple.
[1]

2.2. Research Questions

From the observations in the previous section, we formulate the research questions for the thesis. We aim to achieve a deeper understanding of static backgrounds in detecting moving objects.

- **Do static backgrounds of the training set influence the detection results of a moving object at test time?**
- **If yes, to what extent are static backgrounds exploited for detecting moving objects?**
- **Does masking or removing static background during training make video object detectors robust to static background changes at test time?**

2.3. Outline

The thesis is organized as follows, Chapter.1 is the primary documentation that explains the thesis including the methodology and experiment results. Chapter.3 provides insights into deep learning and object detection. Chapter.4 explains the VID model used, training settings and plots. Chapter.5 presents additional information about the proposed dataset, experiment plots, training, and test settings. Finally, in Chapter.6, we present the additional experiment on masking static objects.

3

Object Detection with Deep Learning

In this chapter, we explore the fundamentals of deep learning and its usage for Object Detection.

3.1. Deep Learning

Deep Learning is a sub-field of machine learning that employs neural networks to imitate the way humans think. Neural networks extract vital information or patterns from data ultimately forming a relationship between input and output. A deep neural network is built by multiple neural network layers. Each neural network layer is composed of neurons. Each neuron has trainable parameters which are learnt by back-propagation. By back-propagation, the parameters are trained and further optimized based on an objective function. A mathematical model of a neuron is shown in Figure.3.1. A neuron receives input, transforms the input and applies an activation function. The commonly used activation functions are hyperbolic tangent, the Sigmoid and the Rectified Linear Unit (ReLU).

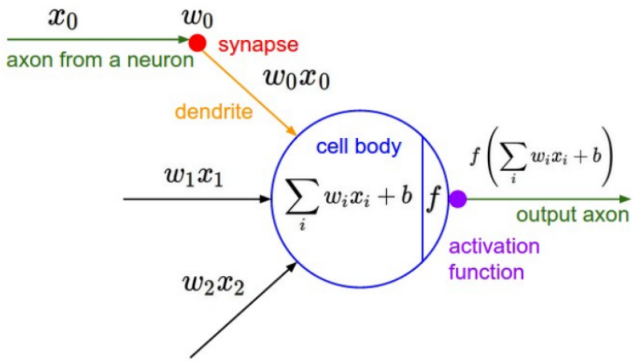


Figure 3.1: Mathematical model of Neuron. The neuron has three inputs x_0, x_1, x_2 with learnable weights w_0, w_1, w_2, b. An affine transformation is applied to the inputs in the cell body. After the affine transformation, an activation function f is applied. The learnt weights represent the strength of connection between neurons.[2]

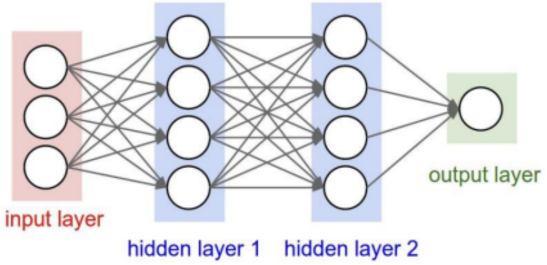


Figure 3.2: A 3-layer Neural Net with an input layer, two hidden layers and output layer.[2]

A general deep neural network is composed of multiple layers connected with each other. They are also known as artificial neural network. Generally, the neural network possesses an input layer,

hidden layer and the output. Neurons of the same layer are not connected. The neural network in Fig.3.2 shows a neural network where each neuron in one layer is connected to every other neuron in the succeeding layer. During training, the neurons establish stronger or weaker connections based on activation in their previous layers. For optimizing the weights between layers algorithms like Stochastic Gradient Descent(SGD) are used. After training, they develop different response to input combinations and provide output in the final layer.

3.1.1. Convolutional Neural Networks(CNNs)

CNNs have been widely used for backbones and detection heads since the introduction of object detection. Similar to neural networks, CNNs comprise four key components namely data, model, objective function and algorithm[3]. For image recognition systems, it is quite straightforward we utilize image data for training. The image data (height x width x number of channels) is fed to the model which extracts relevant features by convolving over every pixel and its neighbours using kernels. Here, the kernel matrix's value determines the image's transformation to feature maps. For example, we show in Fig.3.3 how each convolution kernel can interpret the input image differently. Further, a pooling oper-

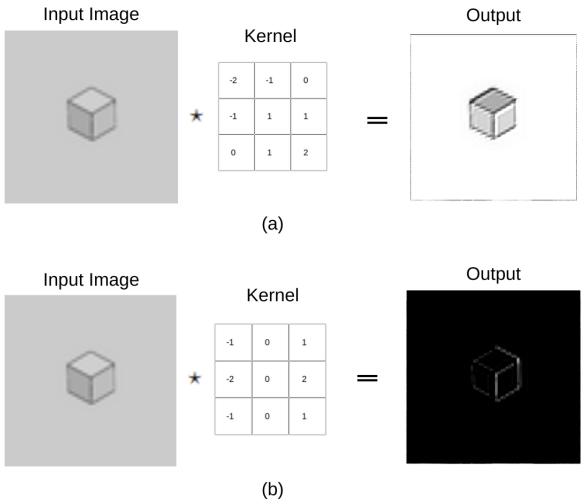


Figure 3.3: An example of image convolution. We use a black and white image for illustration. In (a) We apply a sharpen kernel and visualize a sharper version of the input image as an output. (b) We apply a sobel kernel and visualize differences in adjacent pixels along right direction.

ation is performed at successive convolution layers to reduce the dimension and refines the prominent features in an image or video. Finally, a fully connected layer During training, these kernels are learnt over time with the use of an objective function(commonly referred to as Loss functions) optimized with the help of algorithms like SGD, Adam etc. CNNs have made immense progress and have been quite the mainstream for image recognition with the introduction of VGG[4] and ResNet[5] backbones. For more detailed information about convolutions and how they work refer [6].

3.2. Object Detection

As mentioned earlier in Chapter.2, Object detection involves identifying **what**(classification) and **where** (localization) is the object in a given image or video. Object detection lays the foundation for various applications such as industrial inspection, face detection, automated driving and so on. There exists different type of architectures for object detection. An example of one-shot image object detector[7] is shown in Fig.3.4. Given an input image, the features are extracted with a convolutional network termed as the backbone. Further, the fully connected layers output a classification score accompanied by bounding box coordinates. Based on the type of input, object detection is divided into two parts namely **Image** and **Video** Object Detectors.

The goal of object detection is to detect all instances of classes encountered during test time. Additionally, the trained detector should be able to draw a bounding box around the object boundary. In this thesis, we follow supervised learning, where the image/video along with its ground-truth label is fed to the network for training.

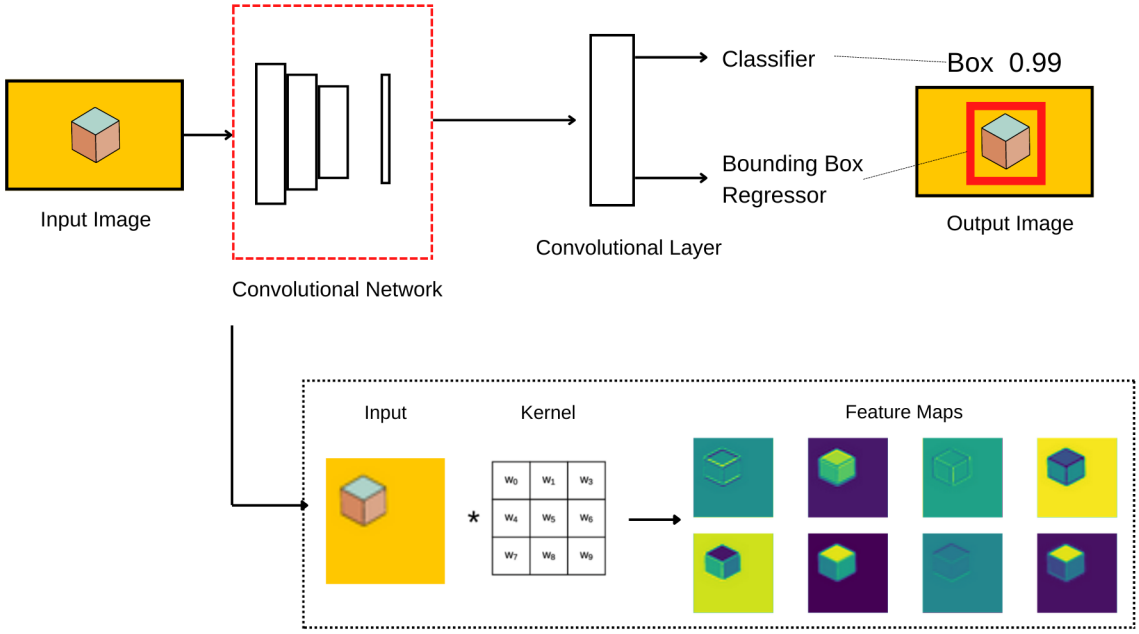


Figure 3.4: An image is given as input to the CNN in which features are extracted. These features are further used to classify and predict bounding box coordinates with the help of an additional fully connected layer. To understand how convolution works, we take a closer look at the block of the Convolutional network (highlighted in red). The image is converted as a tensor and multiplied with a 3x3 kernel with weights $w_0, w_1 ... w_9$ to obtain feature maps. We obtain feature map results in the above figure by implementing a VGG-16[4] pre-trained network on Image-Net dataset[8]

3.2.1. Performance metrics

To evaluate the performance of object detectors, mean Average Precision(mAP)[9] is used. Precision is obtained from Intersection over Union(IoU). IoU is used to identify how much the predicted boundary overlaps with the ground truth. Based on the threshold of IoU, a detection is declared as correct. In our experiments, we use a threshold of 0.5. If IoU overlap is greater than the threshold, we term it as True Positive. On the other hand, if the IoU is less than 0.5 it is declared as False Positive. If an object present in ground truth is not detected then it is classified a False Negative.

$$\begin{aligned} \text{Precision} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \\ &= \frac{\text{True Positive}}{\text{All Observations}} \end{aligned} \tag{3.1}$$

$$\begin{aligned} \text{Recall} &= \frac{\text{False Positive}}{\text{True Positive} + \text{False Positive}} \\ &= \frac{\text{False Positive}}{\text{All Ground Truth}} \end{aligned} \tag{3.2}$$

Based on the above equation, the average precision is computed per class. For the final result, the mean of average precision over all classes is used as the final evaluation. In our results, we report mAP with average over multiple IoU thresholds between 0.5:0.95 with step size of 0.05. This results in computations of AP threshold at ten different IoUs.

4

Video Object Detection

Previously image object detection methods extract features frame by frame. In VIDs, feature extraction occurs in a group of frames. Given the set of features, consistencies are identified using different methods and prominent features are propagated across frames. The main aim of a VID method is to capture temporal information in videos. The VID approach used in this thesis is Sequence Level Semantics Aggregation (SELSA)[10].

4.1. SELSA

The architecture of SELSA is similar to the working of Faster-RCNN[11]. The SELSA modules are added to the existing architecture of Faster-RCNN. The SELSA architecture can be split into three parts namely a Feature Network, Detection Network and the SELSA module. The input set of images is first passed through the feature network which is ResNet-50[12] in our case. The input is also fed to the backbone of the Region Proposal Network(RPN) which is a part of the detection network. The RPN network generates proposals for different scales and aspect ratios. The RPN consists of a softmax layer as a classifier to identify if the proposal boxes are foreground or background. By ranking the boxes according to the classification scores, the final proposals are obtained. Additionally, the RPN consists of the regressor to output the proposal coordinates. Using the proposals from RPN, ROI pooling is applied to feature maps.

After ROI pooling, proposals are passed through two Fully Connected(FC) layers and SELSA modules. The SELSA module first identifies the semantic similarity between the proposals. Further, the features are aggregated to obtain robust features as shown in Figure.4.1. Finally, the proposals are passed to

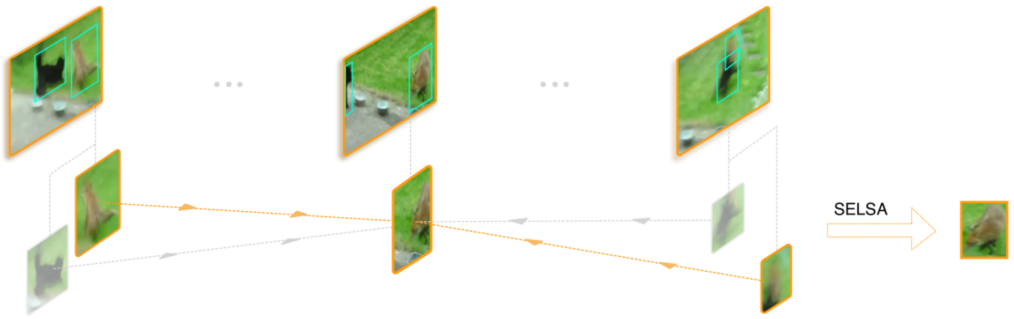


Figure 4.1: Semantic guidance and Feature aggregation in SELSA[10]. Region proposals are extracted from different frames followed by identification of semantic similarities. Finally, the features are aggregated from neighbouring proposals.

two FC layers for object classification and regression. SELSA is jointly trained with four losses namely a binary cross-entropy classification loss for RPN, smooth L1 loss for regression in RPN[11], cross-entropy classification loss and smooth L1 loss for regression[13]. The overall architecture is illustrated in Figure.4.2

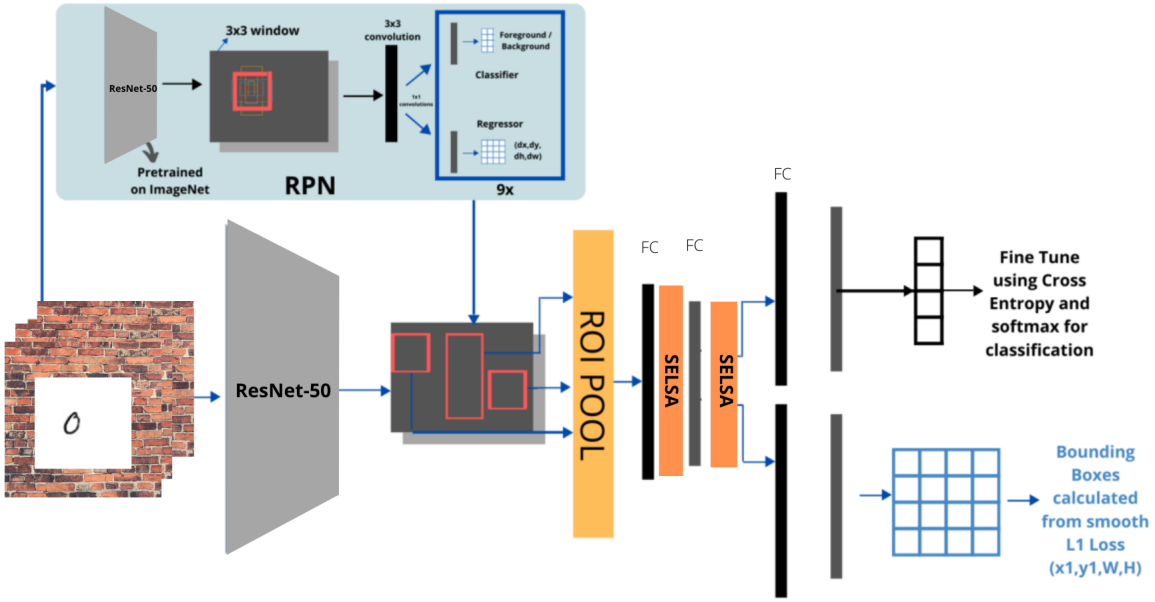


Figure 4.2: Overall architecture of the VID model. The feature network is a ResNet-50 followed by a detection network with RPN. The SELSA modules are inserted between the final fully connected layers.The bounding boxes are output in form of x coordinate, y coordinate, height and width(x1,y1,W,H).

.

4.2. Training details for experiments

Mostly, we adopt the same configuration for all our experiments. We do not employ data augmentation for our experiments. The video frames are normalized with the ImageNet mean and standard deviation values. For training, two random reference images are chosen from a frame range of 9. During, test time, we sample 14 reference frames along with the testing frame. For the optimizer, we use SGD with a initial learning rate of 0.01, momentum of 0.09 and weight decay of 0.0001. The learning rate is divided by 10 at 90k and 270k iterations. We fix the hyper-parameters and train with all variants of the SB-MNIST.

Due to the small size of the dataset, we apply transfer learning by using ImageNet pre-trained weights. We freeze the first layer of the feature extraction network. However, the VID models manage to overfit the datasets as early as 150k iterations. For the Masked-SB-MNIST variant, we increase the extent of transfer learning by freezing two layers. Yet, the results are worse when tested with different static backgrounds. However, setting the weight decay values to 0 improved mAP on an average of 3.2%. training plots for all variants are shown in Figure.4.4.

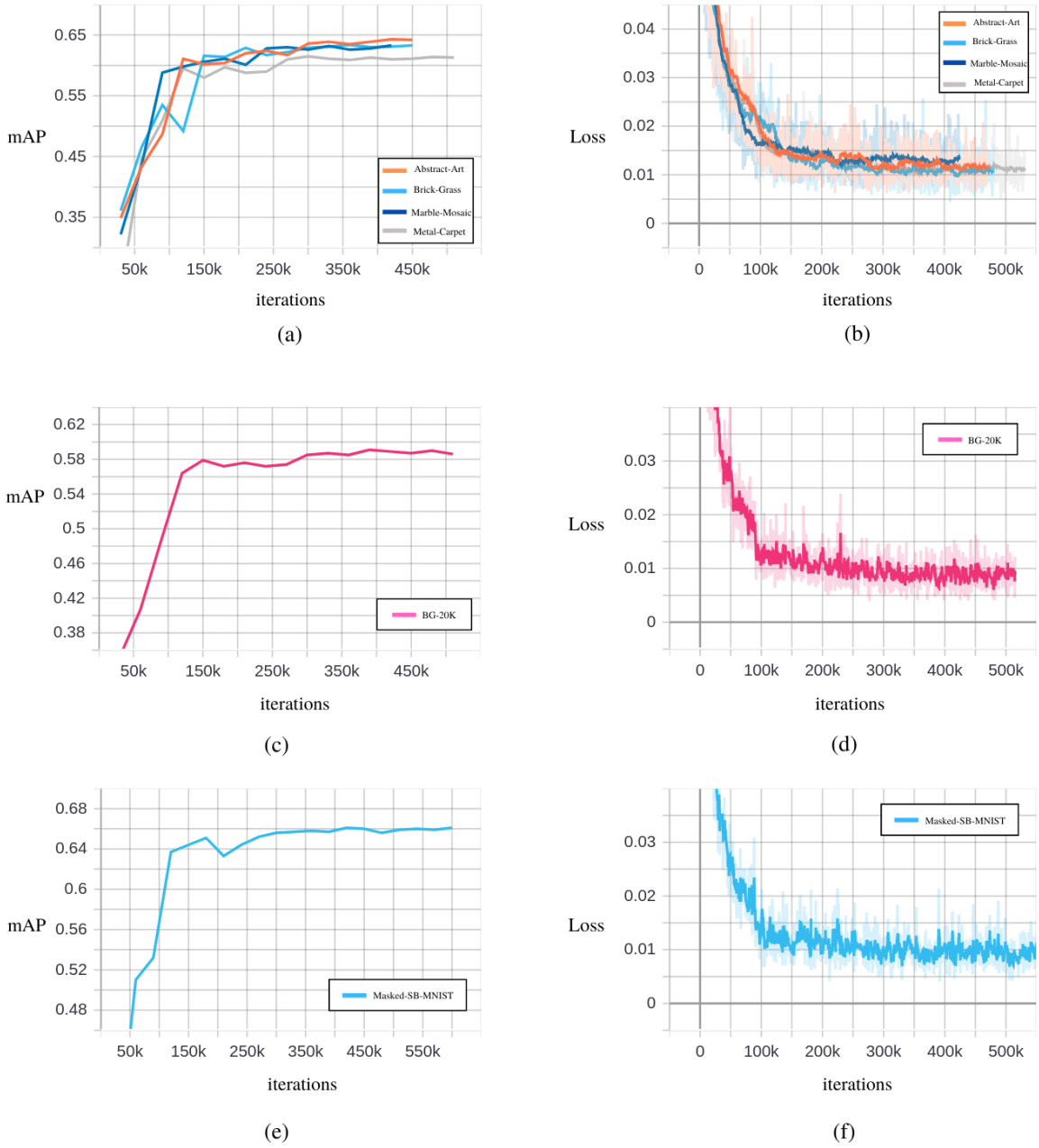


Figure 4.3: Training curves on validation set and combined loss. From top (a) Training curves of VID models with Brick-Grass, Metal-Carpet, Marble-Mosaic and Abstract-Art datasets. As the datasets are small, the models overfit as early as 180k iterations. (b) Combined classification and regression training loss with Brick-Grass, Metal-Carpet, Marble-Mosaic and Abstract-Art datasets. (c) Training curve of VID model with BG-20K dataset. The mAP is values are less than values in (a) as the changing static backgrounds provides regularization. (d) Combined classification and regression training loss with BG-20K dataset. (e) Training curves with Masked-SB-MNIST (f) Combined classification and regression training loss with Masked-SB-MNIST.

.

Figure 4.4: Bounding box detection results of VID model trained on Metal-Carpet variant of Min-SB-MNIST. (a),(b),(c) and (d) are results on Min-SB-MNIST. (e) and (f) are results on Max-SB-MNIST. (h) and (i) are results on No-SB-MNIST.

.

5

SB-MNIST

In Chapter.1, we shortly introduced the datasets for experiments and provide test results. In this chapter, additional information on the SB-MNIST dataset can be found.

5.1. SB-MNIST Dataset

The SB-MNIST is of dimensions 64x64. The moving-object-background is 40x40 and is centred at the frame. The original MNIST[14] digit is 28x28 which is resized to 10x10. The dimensions of an example frame are shown in Figure.5.1. We use the openCV[15] library for image manipulation and dataset creation.

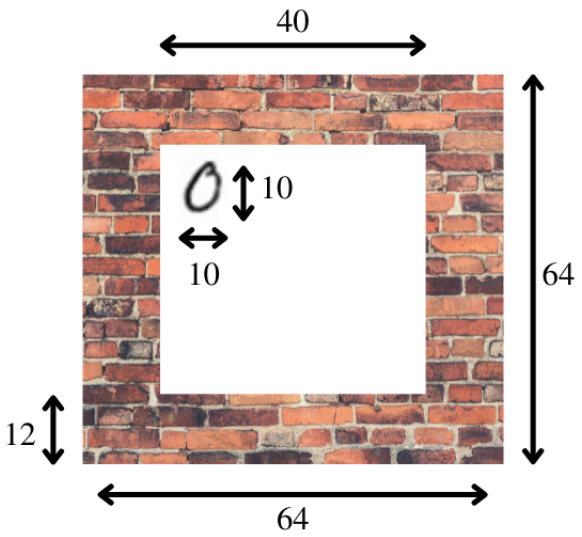


Figure 5.1: Dimensions of a frame in SB-MNIST.

5.1.1. Motion association per class

The Static Background MNIST dataset comprises MNIST digits[14] as moving objects. The moving object's initial location is randomized inside the moving-object-background. Each moving object is associated with a distinct motion pattern as shown in Figure.5.2. When the moving object reaches the ends of moving-object-background, the object bounces back to a random position in the moving-object-background. The digit moves at the speed of one pixel per frame.

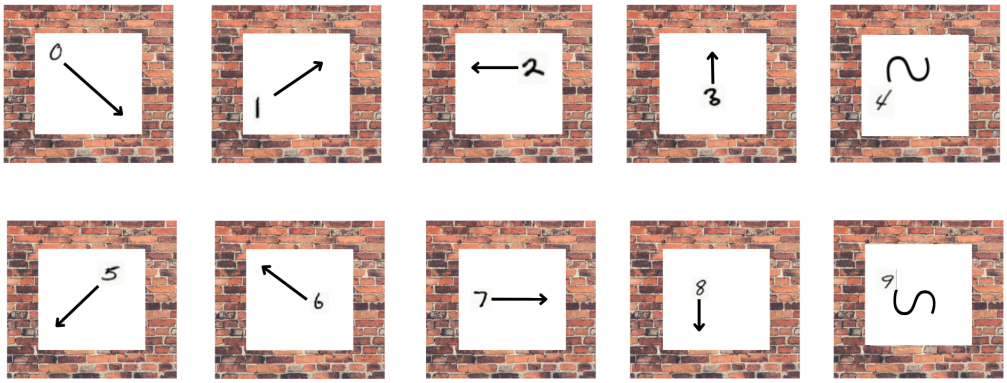


Figure 5.2: Associated motion pattern to each digit from 0 to 9.

5.1.2. Maximum-SB-MNIST

We use several static backgrounds from BG-20K[16] and DTD[17]. For BG-20K, we use static backgrounds from its test set. BG-20K primarily consists of outdoor and indoor scenes. For DTD, we use static backgrounds from 10 texture categories namely stained, stratified, striped, studded, swirly, veined, waffled, woven, wrinkled, and zigzagged. Examples of the static backgrounds used are shown in Figure.5.3 and Figure.5.4.

Figure 5.3: Examples of static backgrounds in BG-20K variant of Max-SB-MNIST

Figure 5.4: Examples of static backgrounds DTD variant of Max-SB-MNIST

6

Additional experiments

To identify the influence of static objects in detecting dynamic objects in a scene, we create two datasets called Mask-Moving and Unmask-Moving based on MNIST digits[14] as shown in Figure.6.1 and Figure.6.2. The dataset comprises 80 videos for train, 20 videos for validation and 20 videos for the test set. The total number of frames for training is 6680. The validation and test sets contain 1840 frames. The length of the videos is 46 and 120 frames in training, while at validation and test time the video length is 46. The dimension of the video frame is 150x150 with the moving object of size 30x30. At each frame, the moving object instance is changed and moves either horizontal, vertical or diagonal. We hypothesize the following for our experiments:

- By masking the static objects during training, the video object detector learns to focus on moving objects and ignore static objects while testing.
- If all static objects in a video are masked during training, the mAP of dynamic objects will improve compared to training with unmasked static objects.

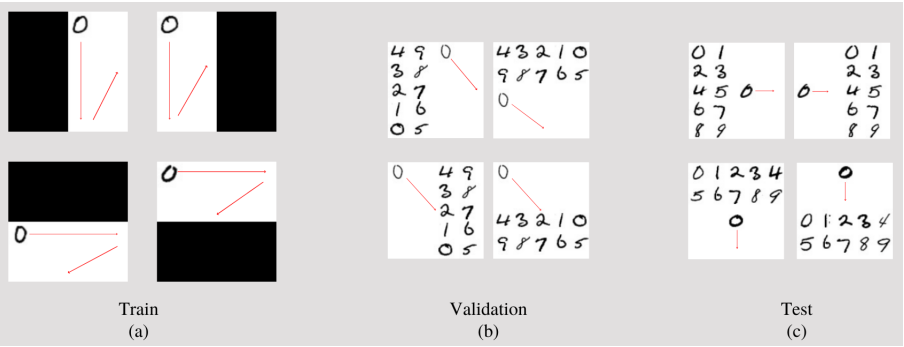


Figure 6.1: Overview of Mask-Moving dataset. (a) Train dataset with masks on left, right, top and bottom. The arrows indicate the direction of movement for the moving objects. (b) Unmasked validation dataset with object moving diagonally. (c) Test set with unmasked static objects. The moving objects move either horizontally or vertically.(b) and (c) are common for both Mask-Moving and Unmask-Moving.

.

6.1. Mask-Moving Vs Unmask-Moving

We trained the SELSA VID model[10] on two datasets. First, we mask the static objects in the scene as shown in Figure.6.1. The model is trained on Mask-Moving. For validation, we do not mask the static objects. During training, the VID model achieves an mAP of 67.6% on the validation set. Similarly, we train the VID model on Unmask-Moving dataset as shown in Figure.6.2. The position of static digits are shuffled for each video. The VID model achieves an mAP of 30.8%. The training curve is shown in Figure.6.3.

After training, we test the VID models on test set which is unmasked. T**he VID model trained on Mask-Moving achieves a better mAP of 67.5%** whereas the VID model trained on Unmask-Moving performs

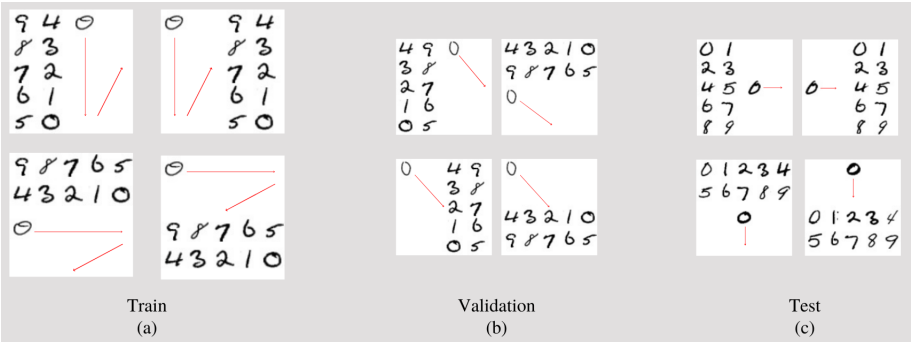


Figure 6.2: Overview of Unmask-Moving dataset.(a) Train dataset with static digits on left, right, top and bottom. The arrows indicate the direction of movement for the moving objects. (b) Unmasked validation dataset with object moving diagonally. (c) Test set with unmasked static objects. The moving objects move either horizontally or vertically.(b) and (c) are common for both Mask-Moving and Unmask-Moving.

.

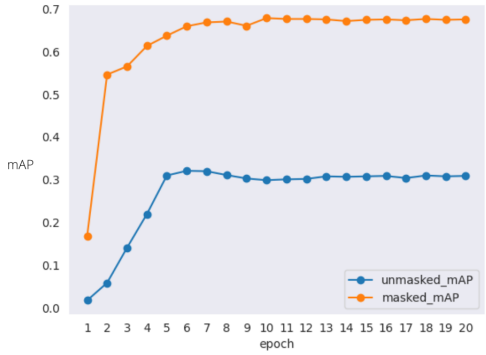


Figure 6.3: The training curves of VID models trained on Mask-Moving and Unmask-Moving. The VID models achieve a greater mAP on Mask-Moving compared to Unmask-Moving.

.

worse by attaining an mAP of 42.3%. Looking at the bounding box detections, it is noticed that the VID model trained on Mask-Moving detects more False positives in static objects. The results help us conclude that masking static objects during training does not help VIDs ignore the static objects. However, Masking is beneficial as it improves the mAP of the moving object. Sample detection results are shown in Figure.6.4.

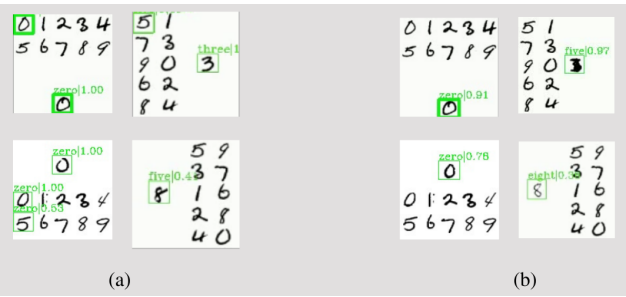


Figure 6.4: Detection results of two VID models tested on Unmasked test set. (a) Results of VID model trained on Mask-Moving and tested on Unmask-Moving test set. False positives occur on the static objects. (b) Results of VID model trained on Unmask-Moving and tested on Unmask-Moving.

.

References

[1] R. L. R. V. Inspections. "Rolls royce trent 500 hp compressor damage," Youtube. (2021), [Online]. Available: https://youtu.be/P4qtModk2FQ.

[2] J. Johnson. "Cs231," Stanford. (2021), [Online]. Available: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture4.pdf.

[3] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *CoRR*, vol. abs/2106.11342, 2021. arXiv: 2106.11342. [Online]. Available: https://arxiv.org/abs/2106.11342.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014. [Online]. Available: http://arxiv.org/abs/1409.1556.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.03385.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Jun. 2015. [Online]. Available: http://arxiv.org/abs/1506.02640.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[9] M. Everingham *et al.*, "The pascal visual object classes (voc) challenge." [Online]. Available: http://www.flickr.com/.

[10] H. Wu, Y. Chen, N. Wang, and Z. Zhang, "Sequence level semantics aggregation for video object detection," Jul. 2019. [Online]. Available: http://arxiv.org/abs/1907.06390.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Jun. 2015. [Online]. Available: http://arxiv.org/abs/1506.01497.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[13] R. Girshick, "Fast r-cnn," Apr. 2015. [Online]. Available: http://arxiv.org/abs/1504.08083.

[14] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[15] O. S. C. Vision. "Background subtraction methods available in opencv," OpenCV. (2021), [Online]. Available: https://docs.opencv.org/3.4/d8/d38/tutorial_bgsegm_bg_subtraction.html.

[16] J. Li, J. Zhang, S. J. Maybank, and D. Tao, "Bridging composite and real: Towards end-to-end deep image matting," *International Journal of Computer Vision*, 2022, ISSN: 1573-1405.

[17] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.