2018

Investigating the Application of Deep Convolutional Neural Networks in Semi-supervised Video Object Segmentation

Jayadeep Sasikumar
Technological University Dublin

Investigating the Application of Deep Convolutional Neural Networks in Semi-supervised Video Object Segmentation

Jayadeep Sasikumar

MSc in Computing (Data Analytics)

Dublin Institute of Technology

A dissertation submitted in partial fulfilment of the requirements of Dublin Institute of Technology for the degree of M.Sc. in Computing (Data Analytics)

2018

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed:

Jayadeep Sasikumar

Date: 4 January, 2019

Abstract

This thesis investigates the different approaches to video object segmentation and the current state-of-the-art in the discipline, focusing on the different deep learning techniques used to solve the problem. The primary contribution of the thesis is the investigation of usefulness of Exponential Linear Units as activation functions for deep convolutional neural architectures trained to perform object semi-supervised segmentation in videos.

Mask R-CNN was chosen as the base convolutional neural architecture, with the view of extending the image segmentation algorithm to videos. Two models were created, one with Rectified Linear Units and the other with Exponential Linear Units as the respective activation functions. The models were instantiated and fine-tuned on the first frame of each sequence on the test dataset before predicting segmentations. This was done to focus on the principal object in the video for segmentation.

Mean Jaccard index was the metric chosen to evaluate the performance of the models. No significant difference was found between the performance of the two models on the test dataset. A qualitative analysis of the performance of the model with ReLU activation functions was conducted with the view of understanding its strengths and weaknesses. The thesis concludes with an overview and a discussion on limitations and recommendations for future work that can be done to extend on the work presented in this thesis.

Key Words: *Computer Vision, Video Object Segmentation, Deep Learning, Convolutional Neural Networks, Rectified Linear Units, Exponential Linear Units.*

Acknowledgements

I would like to express my sincere gratitude towards my supervisor, Robert J. Ross, for his valuable guidance, kind words of encouragement and admirable patience.

Many thanks to Matterport engineering team for making their implementation of Mask R-CNN available online and making it so easy to work with.

Thanks to Greg, for helping me figure things out, and teaching me to pull myself up when I fall.

And thanks to Anjali, for staying by my side during all the difficult times.

Table of contents

Table of figures

Table of tables

1. INTRODUCTION

1.1. Background

Computer Vision has always been one of the more complex among the host of problems that pose challenge to the development of intelligent automata with sensory inputs. Computer vision is the process of automating the deciphering of patterns, the understanding the semantic information about real world objects represented in the visual media (Ballard & Brown, 1982).

Object Segmentation in videos is currently an area of research that has been garnering a lot of attention, primarily due to the sheer importance the process has in a broad spectrum of problems in the domain of Computer Vision in general, and partly due to the recent technological advances – both hardware and software. The objective of object segmentation in videos is similar to that of object segmentation in images, to identify, and delineate one or more objects present in a video. It is a more complex problem than the more traditional problems like *object detection* (classification and localisation of objects), and *semantic segmentation* (grouping similar pixels in a video). Object segmentation (and the more complex *instance segmentation*) can be considered an extension to these classical problems (He, Gkioxari, Dollár, & Girshick, 2017). The identification and segmentation of objects forms the basis for scene understanding, a mandate for the myriad of applications falling under disciplines ranging from traffic monitoring (Cheung & Kamath, 2007; Remagnino et al., 1997), and autonomous driving (Cordts et al., 2016; Ess, Mueller, Grabner, & Gool, 2009), to intelligent photography (Yoon, Jeon, Yoo, Lee, & Kweon, 2015) among various others.

Research on object segmentation, similar to that on many other problems falling under the umbrella of Computer Vision (and outside it), has been given a boost recently due to the significant progress made in *Deep Learning*. Deep Learning is an example of *representational (or feature) learning methods*, a suite of machine learning algorithms that aim to identify representations or features from data that help in detection or classification of the data itself (Bengio, Courville, & Vincent, 2012). This eliminates the necessity of possessing a deep level of domain-specific knowledge before application of the technique to the data, enabling and even encouraging interdisciplinary research. The resurgence of deep learning techniques in the last decade has seen it pushing the existing limits and dominating the state-of-the-art in a multitude of fields (Yann LeCun et al., 2015).

1.2. Research problem

Video object segmentation is fast becoming a fundamental Computer vision problem because of its importance to a multitude of other related problems spread across a wide range of disciplines. With the release of DAVIS 2016 dataset for semi-supervised object segmentation videos, research has been given a boost and the state-of-the-art pushed multiple times over the last couple of years. Most of the state-of-the-art solutions have their approach rooted in deep convolutional neural networks and employ the Rectified Linear Units as activation units in the network architecture. This research attempts to look at an alternative choice for activation units, Exponential Linear Units, which have been proven to perform better in certain contexts when compare to the Rectified Linear Units, in the context of semi-supervised video object segmentation.

Research Question: Can changing the activation unit of a convolutional neural network trained to perform semi-supervised object segmentation in videos from Rectified Linear Unit to Exponential Linear Unit impact the mean Jaccard Index observed for the model?

1.3. Research objectives

The primary objective of this research is to understand how a technique designed to perform object segmentation in images can be extended to videos, and subsequently, to investigate the impact of using a different neuron, the *Exponential Linear Unit (ELU)*, than the one used in most current architectures, *Rectified Linear Unit (ReLU)*. The intuition behind this research is that videos are but collections of images, which when played continuously, produce an illusion of motion due to persistence of vision (Maninis et al., 2017). Also, extension of image object segmentation techniques to video is not an uncommon approach to the problem (Caelles et al., 2016; Khoreva et al., 2016; Maninis et al., 2017).

A secondary objective is to look into the feasibility and specifics of modelling the implicit temporal structure in the videos by adding a recurrent component to the network architecture constructed in this research. This is motivated by the fact that videos are more than just collections of images, they are *sequences* of images – there is a logical, temporal flow of data from one frame to another. This presents an opportunity to view the same problem a bit differently. The modelling of the temporal structure inherent in the videos had been approached in different ways without using a recurrent component (Cheng et al., 2017; Khoreva et al., 2016; F. Li et al., 2013). There are approaches which specifically try to combine the recurrent and convolutional components to perform video object segmentation (Chen et al., 2016; Hu et al., 2018; Valipour et al., 2016).

The null hypothesis is that there is no impact in the mean Jaccard index observed for a convolutional neural network to perform semi-supervised object segmentation in videos when the activation unit of the network is changed from Rectified Linear Unit to Exponential Linear Unit.

1.4. Research methodologies

This research is of secondary, empirical nature and seeks to analyse and study the impact of the addition of a component to a baseline solution by comparing its performance with that of the baseline model. The data for the research is obtained from the DAVIS 2016 challenge. The code written for this experiment is made available, and the performance metrics are measurable – thus, the research is empirical.

1.5. Scope and limitations

The models developed as part of this research were mostly configured with the hyperparameters that were suggested by the authors of the different architectures used in the research, and the ones preset in the Mask R-CNN implementation used as the base for development. Hyperparameter tuning, although could have possibly improved on the observed results, was not explored thoroughly as it was not the focus of this research.

1.6. Organisation of the dissertation

- **Chapter 2** covers the relevant scientific **literature reviewed** for the purpose of this research. It discusses the problem of object segmentation in videos, the different ways the researchers have approached the problem over the years, identifying three broad clusters under which the solutions are classified. The review then zeroes in on the final and the currently most popular approach to the problem, using deep learning techniques and discusses various solutions rooted in deep learning, before concluding by stating the identified literature gaps.
- **Chapter 3** discusses the **design and the methodology** of the research. It starts out with the dataset selection process, briefly describing the different datasets considered for the purpose of this research, the criteria regarded to find an appropriate dataset and the reasoning behind choosing the final dataset. This is followed by a detailed description of the chosen dataset, an explanation of the evaluation criterion adopted by this research and an overview of the design of the research – the design of the experiment and the subsequent analysis and evaluation of the different techniques.
- **Chapter 4** delves into the details of implementing the research described in the third chapter. It discusses the different **implementation** details involved and the choices made during the development phase and explains the motivations behind those decisions. The chapter concludes with the reporting of the **results** obtained in the experiment.
- **Chapter 5** is the **conclusion** of the thesis. The chapter opens by giving an overview of the research conducted and the experimental setup, proceeding to summarise the results obtained and what it means in the context of the research question, and finally concluding by discussing potential future work that could be undertaken to build on the work done during this research.

2. LITERATURE REVIEW AND RELATED WORK

2.1. Object Segmentation in Videos

Computer Vision has been a heavily researched topic in the field of Computer Science for the most parts of at least the past four decades (Ballard & Brown, 1982; Huang, 1996). The significance of Computer Vision has risen primarily due to the exponential growth of the video data in the world. For context, it is forecasted that 82% of all internet traffic by 2022, up from 56% in 2017 (Cisco VNI, 2018). The sheer magnitude of video data makes it near impossible for humans to be able to process it for the various applications, thus arising the necessity for automating the various video processing tasks (Giordano, Murabito, Palazzo, & Spampinato, 2015). The importance of Computer Vision gains in stature when the infinite applications it impacts, spread across various domains, are also taken into consideration – these domains including robotics, autonomous vehicles, augmented reality, human-computer interaction among several others (Brunetti, Buongiorno, Trotta, & Bevilacqua, 2018).

2.2 Different approaches

Object segmentation in videos is a problem of correctly classifying the pixels belonging to (an) object(s) in the video with the view of separating it from the background (Perazzi et al., 2016). It is one of the fundamental tasks for many of the diverse applications of Computer Vision, ranging from pedestrian detection and tracking (Brunetti, Buongiorno, Trotta, & Bevilacqua, 2018), behaviour understanding and event detection (Giordano, Murabito, Palazzo, & Spampinato, 2015) to temporal stabilisation of three-dimensional videos (Erdem, Ernst, Redert, & Hendriks, 2005). All of these applications would require being able to identify the pixels of a frame as part of an object and maintain the identification through out the length of the video – which is object segmentation.

2.2.1 Background Subtraction

There have been various approaches towards solving the problem of object segmentation in videos. One of the more popular approaches during the initial times involved modelling for the background in the video, the idea then to identify the background across the frames in the video and subtracting the proposed background segmentation from each frame, thus coming up with the actual objects in the video. This approach is called Background Subtraction (BS). There have been a host of BS methods over the years, the work over the years laying out the foundation for the future research by identifying some of the core challenges involved in the task. Toyama, Krumm, et al. (1999) proposed the then state-of-the-art approach for object segmentation in videos, Wallflower, an algorithm they explained as based on a concept called Background Maintenance, itself grounded on Background Subtraction. They identified and discussed on some of the common challenges faced in the task, changes in illumination, moving objects and presence of shadows among them.

Background Modelling techniques could further be classified into recursive and non-recursive methods, based on their use of a buffer for the background segmentation. Non-recursive techniques employ a sliding-window strategy for the estimation of background from a scene. They use a frame buffer to keep track of the previous frames to learn the temporal variation of the pixels over those frames and thus aid in the prediction of the background in the next frame. Median filtering (Cutler & Davis, 1998; Zhou & Aggarwal, 2001), medoid filtering (Cucchiara, Grana, Piccardi, & Prati, 2003), linear predictive filter (Toyama, Krumm, et al., 1999) all are instances of non-recursive techniques. On the other hand, recursive techniques are the ones that did not use a buffer to maintain the information of the previous frames, these updated the model for the background recursively at each frame. Approximated median filters (McFarlane & Schofield, 1995; Remagnino et al., 1997) and Kalman filters (Heikkilä & Silvén, 1999; Wren, Azarbayejani, Darrell, & Pentland, 1997; Zhang & Ding, 2012) are some popular examples of recursive techniques. While the recursive algorithms take up less storage when compared to the non-recursive ones, recursive algorithms bear with them the risk of propagating any error in the background model over the frames (Cheung & Kamath,2007).

Friedman & Russell (1997) proposed an upgrade to Kalman filter for object segmentation and tracking, Mixture of Gaussians (MoG). The key difference between the MoG method and Kalman filters was the number of Gaussian distributions tracked – while Kalman filter tracked a single Gaussian distribution, MoG method tracked multiple Gaussian distributions, maintaining a density distribution per pixel. Stauffer & Grimson (2000), developed a stable, robust outdoor object tracking system that was, to a degree, capable of coping with changes in illumination, noisy background, and long-term changes in the complex scene using an MoG method. Cheung & Kamath (2007) discuss a host of Background Subtraction techniques used in processing complex scenes in the context of urban traffic. They recognise the necessity of a BS model to be robust in handling various complexities like illumination changes and non-stationary background components and found the model developed by Stauffer & Grimson (2000) to be the best-performing. They concluded that even though the Mixture of Gaussians method was preforming the best among the various techniques they surveyed, it is significantly complex (a large number of sensitive parameters that required attentive tuning) than the much less computationally complex approximated median filter which compares in the performance as well. Besides, they found the MoG methods to be extremely susceptible to sudden changes in global illumination, thus making it a far from perfect solution. Background Subtraction techniques have been thoroughly examined and explained in literature, with many survey papers deeply researching the topic (Piccardi, 2004; Bouwmans, El Baf, & Vachon, 2008; Bouwmans, 2009; Bouwmans, Baf, & Vachon, 2010).

2.2.2 Graph-based techniques

Another popular suite of techniques that are used in object segmentation in videos employ a graph-based approach, the key characteristic of these techniques being the modelling of a video as a spatio-temporal graph. In contrast to the Background Subtraction techniques, the graph-based techniques attempt at modelling the objects in the foreground rather than the background. The graph-based approaches have been used to tackle the problem of unsupervised object segmentation in videos with a good degree of success. This is tougher than the supervised version because of a lack of prior definition of any object in the video. This unavailability of information leads to a low-level grouping of similar pixels without any semantic value attached to it, called over-segmentation (Lee, Kim, & Grauman, 2011).

Grundmann, Kwatra, Han, & Essa (2010) introduced an approach in which they built hierarchical trees composed of over-segmented spatio-temporal regions of the spatio-temporal graph representing the entire video. Besides the creation of the tree, they employed a dense optical flow to prune the tree in order to try and ensure that any constituent temporal connections are of high quality, resulting in a high-quality solution to the problem of long-term temporal coherence in video object segmentation. They propose their method as a preprocessing step for other segmentation techniques that want to model the temporal component of videos.

Lee, Kim, & Grauman (2011) uses another unsupervised graph-based approach in which they try to move past the over-segmentation technique to the automation of discovery of a set of key segments that can be used to explicitly model object-like motion and (temporal and spatial) persistence. They use a region proposal technique, originally proposed by Endres & Hoiem (2010), to come up with candidate object proposals, rank the proposals on their static appearance and global motion tendencies – an attempt at modelling the proposed object's centrality to the video. The top ranked regions are then checked for matching features across frames to create object-wise likelihood maps which in turn are used in binary pixel-wise classification, achieving global segmentation of the scene.

Li et al. (2013) developed a graph-based technique which identified multiple segment tracks from a pool of proposed segments, for each of which a global appearance model is trained to learn incrementally. A similar incremental learning approach has been adopted by Babenko, Yang, & Belongie (2011) for object tracking in videos to good effect. The entire video is used to train all of the individual models created for the proposed segment tracks and this allows for efficient tracking, which are further optimised using a composite statistical algorithm which makes use of the global appearance models. They reported state-of-the-art performance on a dataset they released along with the paper, the SegTrack v2 dataset. This dataset has since been used to evaluate segmentation models and was considered for the purpose of this research as well.

Graph-based approaches have been used in semi-supervised environments as well. The semi-supervised task involves the provision of segmentation for some of the frames, possibly just the first frame of the video sequence(Bai & Sapiro, 2007; Price, Morse, & Cohen, 2009). The semi-supervised algorithms make the training interactive, allowing the user to annotate the foreground objects on the requisite frames of a previously unseen video sequence. For instance, some techniques propagate the user-annotated segments across the video to produce good results (Price et al., 2009; Fan, Zhong, Lischinski, Cohen-Or, & Chen, 2015).Yuen, Russell, Liu, & Torralba (2009) developed an online open-access system which lets users interact with images and annotate them, thus effectively contributing with a database comprised of a wide range of video sequences.

Some other techniques made use of in other graph-based approaches include higher-order Markov random fields (Ren & Malik, 2007; Babenko, Yang, & Belongie, 2011; Tsai, Flagg, Nakazawa, & Rehg, 2012), and variational approximations (Badrinarayanan, Budvytis, & Cipolla, 2013; Unger, Werlberger, Pock, & Bischof, 2012) among others. Readers are recommended to read on graph-based approaches to solving image segmentation to get a better picture on the construction of spatial graphs (Camilus & V K, 2012; Peng, Zhang, & Zhang, 2013; Wang, 2015). One of the main caveats with the graph-based approaches is that the construction of spatio-temporal graphs remains an extremely intensive task computationally, rendering it both expensive and slow, rendering it difficult to be used in real-time applications (Hu, Huang, & Schwing, 2018).

2.2.3. Deep Learning techniques

The next major approach to solving the problem of object segmentation in videos is by employing Deep Learning. This is currently emerging as the most popular method owing to the fact that the current state-of-the-art is dominated by various Deep Learning approaches (Hu, Huang, & Schwing, 2018). In fact, Deep Learning methods have pushed the boundaries and improved on the state-of-the-art on other areas such as speech recognition and drug discovery among others (LeCun, Bengio, & Hinton, 2015). Although Deep Learning is not a new concept and has been a major topic of research for the best parts of the last three decades (Y. LeCun et al., 1989), the unavailability of capable hardware was a major hurdle to the advances in the area; however, recent progress in hardware, along with the software and algorithmic advancements, has helped research in Deep Learning a lot, thus impacting the wide range of domains mentioned above (LeCun, Bengio, & Hinton, 2015).

Deep Learning based techniques for video object segmentation more often than not extend on image object segmentation techniques, owing to the fact that a video sequence can be considered as a collection of images (Garcia-Garcia et al., 2018). That the Deep Learning techniques which applied an image segmentation algorithm on a frame-to-frame basis performed close to the state-of-the-art makes it compelling to review some of the image segmentation algorithms in this chapter.

2.2.3.1 Image segmentation algorithms based on Deep Learning

Convolutional Neural Networks (CNN) have been the most used Deep Learning architecture in segmentation tasks due to its tremendous ability of learning spatial features (LeCun et al., 1989; Krizhevsky, Sutskever, & Hinton, 2012). The convolutions / sliding-window taken by CNNs make them excellent in preserving spatial patterns (LeCun, Bengio, & Hinton, 2015). Garcia-Garcia et al. (2018) presents segmentation as a product of natural evolution of simpler and coarser problems, having their origins in classification, like image classification and the finer detection and localisation problems. Segmentation, classification of each pixel into background or foreground, thus can be considered the finest in this scale of problems, a natural extension to the mentioned classical problems (He, Gkioxari, Dollár, & Girshick, 2017). The section proceeds to examine some of the seminal deep neural network architectures.

While CNNs have been used in image classification since the nineteen eighties (LeCun et al., 1989), it is only recently that the advancements in Computer Vision have been impacted greatly by them. This is mostly due to the advent of deeper network architectures. AlexNet, introduced by Krizhevsky, Sutskever, & Hinton (2012) was a pioneer effort in image classification, improving on the then state-of-the-art by more than ten percentage points in accuracy. The architecture, while not very deep when considered by today's standards (He, Zhang, Ren, & Sun, 2015; Xie, Girshick, Dollár, Tu, & He, 2016), was complex enough at the time that it was implemented by splitting it into two and running it in two GPUs. This shows how much the domain of Deep Learning has progressed in the space of the past eight years. Simonyan & Zisserman (2014) of Visual Geometry Group from the University of Oxford improved on this with their proposal of VGG-16, making it easier to train yet deeper models. Szegedy et al. (2014), in their model GoogLeNet, introduced a new building block to the neural architectures, inception module, which rethought the way of stacking convolutional layers, making the parallel computation of a Network in Network (NiN) layer, a pooling layer, and two convolutional layers possible. GoogLeNet reduced considerably on the number of parameters required to be learned, bringing down both the memory required and the computational expense in the process.

However, the results were observed to saturate after a certain depth (He & Sun, 2014; Srivastava, Greff, & Schmidhuber, 2015), rendering the building of deeper networks not very useful. He et al. (2015) introduced the concept of residual blocks in their seminal ResNet (short for Residual Network) architecture to address this problem. They solved the problem of saturation of results by introducing identity layers (or skip layers), layers capable of copying their inputs to the immediate next layer, the intuition being that a layer gets to learn not only from what its immediate predecessor outputs, but also from what the predecessor layer had available to learn from. This makes the propagation of features deeper into the layers, thus combating the problem of vanishing gradients as well. The simple nature of the solution meant a further reduction in complexity in training, thus enabling the training of much deeper networks and making them faster too. ResNet, introduced in 2015, won the ILSVRC-2016 challenge (Russakovsky et al., 2014) recording an accuracy of 96.4%. This was more than eleven percentage points than the accuracy recorded by AlexNet (84.6%) when winning ILSVRC-2012 challenge. This again reiterates the progress made by the Deep Learning architectures in the area in the relatively short time span of four years.

Another seminal work by Long, Shelhamer, & Darrell (2014) introduces a Fully Convolutional Network (FCN), replacing the fully connected layers at the end (that was a characteristic of most prominent architectures at the time) with further convolutional layers. FCNs proved the feasibility of using convolutional layers throughout for problems of a similar nature. The replacement of fully connected layers with convolutional layers meant further reduction in the number of parameters to be learned. The feature maps produced by the final convolutional layers are then upsampled by applying fractionally strided convolutions, commonly referred to as deconvolution layers (Zeiler & Fergus, 2013; Zeiler, Taylor, & Fergus, 2011). FCN has considerably improved the performance of their traditional variants and as such, is currently the most popular approach adopted by the researchers trying to improve on the performance – most of the current state-of-the-art feature FCNs. These networks have been so important in that these are considered the building blocks for the newer and improved solutions (Garcia-Garcia et al., 2018).

The discussed progress in image classification has evidently been helped on by the presence of various online challenges focusing on the task, the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) conducted by the ImageNet project (Russakovsky et al., 2014) one of the most popular ones – the winner of which has de facto been considered the state-of-the-art over the years. Likewise, the Pascal Visual Object Classes (Pascal VOC) annual challenge (Everingham, Gool, Williams, Winn, &

Zisserman, 2010) has also been instrumental by providing an exemplary dataset for image classification, object detection and segmentation. The Densely-Annotated VIdeo Segmentation (DAVIS) annual challenge (Caelles et al., 2018; Perazzi et al., 2016; Pont-Tuset et al., 2017) is regarded similarly as a benchmark dataset in the domain of video object segmentation. From 2017 onwards, the datasets released as part of the respective DAVIS challenges have provided separate segmentations for different objects in the frame, thus making it suitable for training instance segmentation models as well – the process of not only segmenting objects, but identifying the different instances of each object as well.

While most of the recent techniques employ FCNs to achieve segmentation in images, the constituent components vary. Classical networks like VGG-16 or ResNets sans their fully connected layers are used to extract the spatial features and then a deconvolution network is connected to this part to upsample the resultant feature map. This is akin to an encoder-decoder architecture, with the extraction of spatial features and generation of smaller sized feature maps being the encoding component and the subsequent upsampling of these low resolution feature maps to pixel-accurate segmentations being the decoding component. The final layer in the architecture could still be a softmax classification layer. SegNet presented by Badrinarayanan, Kendall, & Cipolla (2015), and U-Net presented by Ronneberger, Fischer, & Brox (2015) are two examples of this encoder-decoder architecture.

The more conventional approach of considering the problem of segmentation as a pixel-wise classification also has been adopted in research and commendable progress made on the area. Gu (2009) discussed about the importance of features in identifying regions of potential interest in a complex scene. A unified framework for object detection, classification and segmentation was presented. Uijlings, Sande, Gevers, & Smeulders (2013) successfully extended on this paradigm of using regions for recognition and introduced the Selective Search algorithm for object detection. Girshick, Donahue, Darrell, & Malik (2013) presented a method called Regions with CNN features (R-CNN). R-CNN used the Selective Search algorithm for generating proposals of regions of interest (RoI) in the image. These proposals were then classified into objects and the regions refined using a linear regression model. The R-CNN architecture was improved remarkably on its running time by its successor, Fast R-CNN (Girshick, 2015). This was achieved by the unification of the different constituent networks in the R-CNN architecture, and by the adoption of a technique called the RoIPool (Region of Interest Pooling), which eliminated the need for running a forward pass per proposal and reduced it to one single forward pass for all the proposals. The Fast R-CNN architecture was further sped up by removing the use of the relatively slower Selective Search algorithm (Uijlings, Sande, Gevers, & Smeulders, 2013). The deep convolutional network that performed the feature extraction was used for region proposal as well. The resultant architecture was dubbed Faster R-CNN (S. Ren, He, Girshick, & Sun, 2015). He et al. (2017) extended on this architecture and presented an architecture aimed at solving the task of instance segmentation. The RoIPool technique was replaced by a novel technique proposed called RoIAlign, which helped preserve the exact spatial locations of features and fixed a misalignment caused by RoIPool. This architecture, Mask R-CNN, improved on the state-of-the-art for instance segmentation in images and went on to win the prestigious Marr Prize for the year 2017, annually awarded to the best paper by International Conference on Computer Vision (ICCV).

2.2.3.2 Deep Learning for video object segmentation

As discussed before, many of the video object segmentation techniques extend on existing image segmentation techniques, the intuition behind doing so being the fact that videos are but collection of images (the constituent frames). However, videos should be considered a sequence of images (frames) rather than just a collection of them (Maninis et al., 2017). This means that the temporal information contained in the videos can be used to aid in the process of video-related Computer Vision tasks, including object segmentation or instance segmentation. As such, there have been various approaches to modelling the inherent temporal component for the same.

MaskTrack, an architecture introduced by Khoreva, Perazzi, Benenson, Schiele, & Sorkine-Hornung (2016), try to model the temporal flow by passing the segmentation of the previous frame along with the RGB channels of the current frame as input to the model. This approach has been the basis for a host of solutions since. The top three architectures in the DAVIS 2017 challenge all have MaskTrack as their base technique – Video Object Segmentation with Re-identification (VS- ReID) model (X. Li et al., 2017), LucidTracker (Khoreva, Benenson, Ilg, Brox, & Schiele, 2017), and Instance Re-Identification Flow (IRIF) method (Le et al., 2017) are the three papers that came first, second and third respectively in the challenge. There have been different approaches to this though. Jain, Xiong, & Grauman (2017), have formulated the problem as a structured prediction problem and tried to solve it by their approach, FusionSeg, by implementing parallel networks to capture the motion and appearance of the objects and then unify it for the final segmentation of the various objects. Hu, Huang, & Schwing (2018) introduce a novel approach called MaskRNN that has a base very similar to the MaskTrack model and has an additional recurrent component in it to model the sequential nature of the temporal information contained in the video.

The primary focus of this paper is however on techniques that do not model the temporal information in the videos. One-Shot Video Object Segmentation (Caelles et al., 2016), shortened as OSVOS, tries to run an efficient image object segmentation technique on the independent frames. This technique has been extremely influential as well and inspired many other improved approaches to the problem since then (Maninis et al., 2017; Newswanger, 2017; Shaban et al., 2017; Voigtlaender & Leibe, 2017). OSVOS tackles the semi-supervised version of the object detection problem, where the segmentation of the first frame of a video is available. OSVOS has a modular architecture that starts with a deep FCN-based architecture pre-trained on a large dataset (like ImageNet or MS-COCO) that acts as the base network. This network is then re-trained on the particular dataset, thus attuning the model more specifically to the problem at hand. The final step is the key in this approach. This involves fine-tuning the entire network using the first frame and its ground truth. This step makes the model tailored to the specific video sequence at hand, and is rerun for each video sequence to segment. The original OSVOS architecture has a VGG-16 as its base network, but this is a customisable component in the modular architecture – any classical deep convolutional network converted to an FCN could be plugged in as the base network.

2.3 Gaps in literature

2.3.1 Can Rectified Linear Unit be replaced?

From the literature review, it was observed that most of the relevant architectures employ Rectified Linear Units (ReLUs) as the activation function (Caelles et al., 2016; He et al., 2017; Maninis et al., 2017). That ReLU is a non-saturated activation function is a clear advantage over the saturated functions like sigmoid activation function or the hyperbolic tangent function because saturated functions are susceptible to exploding gradient and vanishing gradient problems when the architecture gets deeper and are slower to converge when compared to ReLU (Xu, Wang, Chen, & Li, 2015). But, ReLUs have a problem that they can 'die' off. At larger learning rates, a ReLU unit can be updated in a way that it will not activate irrespective of the data and will always output zero from then on. This forces the use of smaller learning rates. While not a big problem in itself as the problem rarely occurs when smaller learning rates are used, it would be interesting to study how the alternative non-saturated functions that are immune to this problem would perform.

Exponential Linear Unit, shortened as ELU (Clevert, Unterthiner, & Hochreiter, 2015), counters this problem by not cutting off the negative component of the function completely like ReLU. Also, ELU is proven to speed up convergence and perform comparably to ReLU, and even surpass ReLU and Leaky ReLU (Maas, 2013) at least in generalisation performance in certain contexts. ELU introduces another tuneable hyperparameter into picture – α, the negative gradient coefficient. Pedamonti (2018) conducts an experiment to compare how ReLU and ELU perform as activation functions in a CNN tuned to a relatively simple image classification task and reported that ELUs performed marginally better than ReLUs in that specific context. ELUs have been used in the context of object classification and segmentation in images in the context of aerial imagery (Panboonyuen, Jitkajornwanich, Lawawirojwong, Srestasathiern, & Vateekul, 2017) and found that the results compared with the state-of-the-art. Panboonyuen, Vateekul, Jitkajornwanich, & Lawawirojwong (2018) introduced ELU as the activation function in a Convolution - deconvolution (encoder - decoder) architecture in the same context, road segmentation from aerial imagery. The impact the introduction of ELU as an activation instead of the now conventional ReLU can have in the context of video object segmentation would be interesting to explore.

2.3.2 A recurrent neural component to model the sequential nature of videos.

While there are many different approaches to modelling the temporal nature of the videos in object segmentation problem, recurrent neural networks (RNN) are very rarely used to address it. RNNs are proven to perform excellently when it comes to modelling sequential data (Karpathy, Johnson, & Fei-Fei, 2015; Yin, Kann, Yu, & Schütze, 2017). As discussed, videos are inherently sequential data as it has a logical temporal structure and flow to it. While Hu et al. (2018) uses a recurrent component in MaskRNN, it is not clearly described how the recurrent component helped in modelling the problem. Some other approaches did talk about using a recurrent component in similar contexts (Chen, Yang, Zhang, Alber, & Chen, 2016; Valipour, Siam, Jagersand, & Ray, 2016), but these models do not use the state-of-the-art convolutional architectures. A study could be undertaken to see how the addition of a recurrent component to a current Deep ConvNet architecture would perform and compare against the state-of-the-art.

2.4 Summary

In this chapter, the literature reviewed for this research has been discussed. The review started with how the solutions to solving the problem of object segmentation have progressed over time. During the process, the solutions reviewed were grouped under three broad categories – background subtraction, graph-based approaches and finally, techniques based on deep learning. The focus of the review was on the deep learning techniques as the research also focuses particularly on deep learning techniques rather than general approaches. Two gaps were identified in the literature and discussed – this research would undertake to investigate the first of the gaps, how the application of Exponential Linear Units (ELUs) could potentially impact the solution.

3. DESIGN AND METHODOLOGY

3.1. Introduction

This chapter aims to explain in detail how the experiment was set up to answer the research question that has been established prior. Firstly, an overview of the different datasets considered for the research is given and the thought process behind the selection of the final dataset explained. That is succeeded by an overview of the design of the experiment. The chapter concludes by defining the evaluation metric to be used in the experiment to accept or reject the null hypothesis.

3.2. Datasets considered

Segmentation tasks in video have been gaining traction in the computer vision community in the recent years and this has been fuelled by the accessibility of some very good datasets. Some of the popular datasets used in the video segmentation domain were considered for this research. Three of the considered datasets offered pixel-accurate ground truths, which was a key criterion for the dataset selection – SegTrack v2 dataset (Li, Kim, Humayun, Tsai, & Rehg, 2013), Freiburg-Berkeley Motion Segmentation (FBMS - 59) dataset (Ochs, Malik, & Brox, 2014), and Densely Annotated Video Segmentation (DAVIS) dataset (Perazzi et al., 2016). These three shortlisted datasets are summarised in Table 3.1.

Dataset	# of video sequences	# of frames	# of objects per frame	# of attributes annotated	Pixel-accurate segmentation	Image resolution
SegTrack v2	14	976	1 to 6	6	Yes	Varying
FBMS - 59	59	720	Multiple	NA	Yes, but not complete	Varying
DAVIS 2016	50	3455	1	15	Yes	Consistent

Table 3.1 A comparison of the different datasets considered

FBMS - 59 dataset was an extension of the original Berkeley Motion Segmentation dataset, referred to as the BMS - 26 dataset (Brox & Malik, 2010). BMS - 26 dataset itself is made up of 12 video sequences from the Hopkins 155 dataset (Tron & Vidal, 2007). The images of the dataset are mostly devoid of common challenges faced in the video segmentation problem, like occlusion, fast motion, etc. Further, while the available ground-truths are pixel-accurate, the segmentations are not provided for all of the 720 frames. The 26 video sequences inherited from the BMS - 26 dataset has pixel-accurate ground-truths for all the frames, but the added-on 33 sequences provide ground-truths for one in twenty frames. The images, while spatially dense in their nature, are less diverse than the other datasets in that the number of objects in the images, with only animals, cars and people as the classes provided. Although, the video sequences are of very short lengths, averaging 12.2 frames per sequence. Also, the constituent video sequences are of varying resolutions, thus adding an extra overhead in an implementation of the solution due to handling of this variance.

The SegTrack v2 dataset, on the other hand, is comprised of video sequences of longer durations (average is 69.71 frames per video sequence). Besides the pixel - accurate ground truths, extra annotations are provided indicating some of the challenges posed in the respective video sequence. Each of the sequence can have one or more of these six challenges (motion blur, appearance change, complex deformation, occlusion, slow motion, interacting objects). This extra annotation makes possible a better qualitative evaluation of any proposed solution. However, that the dataset consists only 14 video sequences and that the image resolution of the frames of the sequences are varying makes it less desirous for this research.

DAVIS 2016 dataset better suits this research in that it is comparable to the FBMS - 59 dataset in size, having 50 video sequences while not compromising on the length of the individual video sequences. DAVIS 2016 dataset has an average of 69.1 frames per video sequence (comparable to the SegTrack v2 dataset). The additional annotation of video sequences with the challenges posed in them is provided in this dataset as well, only in a more detailed fashion. Each video sequence in the dataset has one or more of the fifteen challenges annotated to it. A summary of these challenges as given by Perazzi et al. (2016) is provided in Table 3.2. The video sequences in the dataset maintains a consistent resolution of 854 X 480. Also, each frame in the video sequences in this dataset has only one primary object to be identified and segmented.

This makes it simpler for the research to better focus on its purpose of investigating how an image segmentation algorithm could be extended to solve a video segmentation problem. Also, DAVIS dataset has grown in its magnitude and importance since 2016 and the maintainers of the dataset have been hosting an annual video segmentation challenge since 2017. The dataset has grown in size, and video sequences with multiple objects have been introduced to the dataset. This is another positive because this research can be built upon and extended to the future editions of the dataset with considerable ease. Thus, the DAVIS 2016 dataset was chosen for the purpose of this research.

3.3. Dataset - DAVIS 2016

The dataset selected for the task is the Densely Annotated Video Segmentation (DAVIS) Dataset (Perazzi et al., 2016). This is a high quality dataset with 50 short length video sequences (3455 frames in all) captured at 24fps. Pixel perfect annotations of the objects in each sequence are also made available. The clips in the dataset contain one object or two spatially connected objects which are considered as a single object.

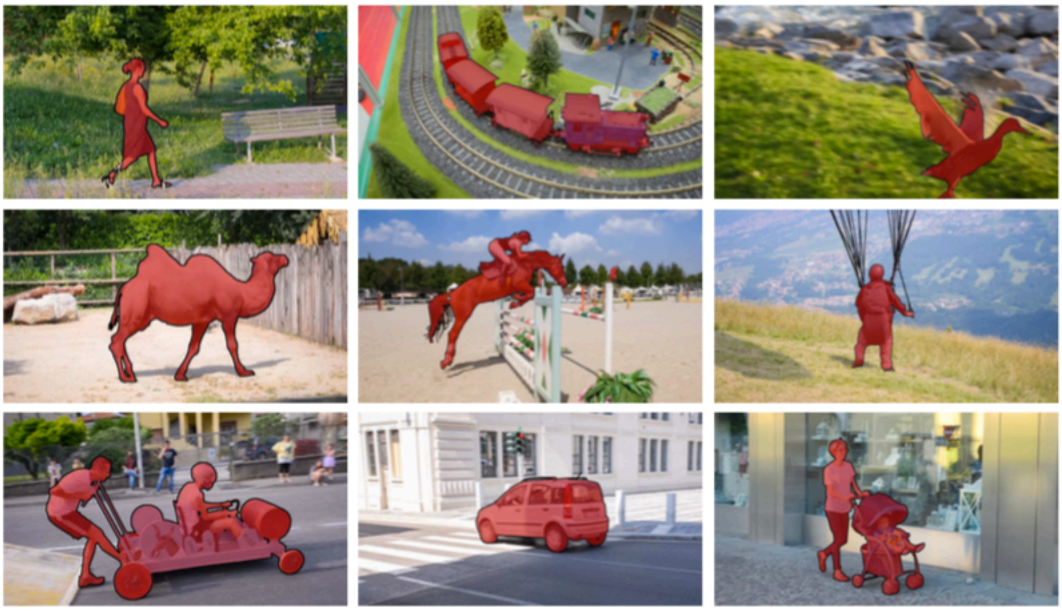

Figure 3.1 Sample frames from the DAVIS 2016 dataset (Perazzi et al., 2016)

This dataset is simple enough to work with because of the single annotation per image, yet complex enough in that most of the major commonly issues faced in the task of object segmentation are present in this dataset. Table 3.1 gives a list of all these common problems as identified by Perazzi et al. (2016).

ID	Description
AC	*Appearance Change*. Noticeable appearance variation, due to illumination and relative camera-object rotation.
BC	*Background clutter.* The background and foreground regions around the object boundaries have similar colours.
CS	*Camera-Shake*. Footage displays non-negligible vibrations.
DB	*Dynamic Background*. Background regions move or deform.
DEF	*Deformation*. Object undergoes complex, non-rigid deformations.
EA	Edge Ambiguity. Unreliable edge detection. The average ground truth edge probability is smaller than 0.5.
FM	*Fast Motion*. The average per-frame object motion, computed as centroid's Euclidean distance is larger than 20 pixels.
HO	*Heterogeneous Object.* Object regions have distinct colours.
IO	*Interacting Objects*. The target object is an ensemble of multiple, spatially-connected objects (eg. mother with stroller)
LR	*Low Resolution*. The ratio between the average object BB area and the image area is smaller than 0.1.
MB	*Motion Blur*. Object has fuzzy boundaries due to fast motion.
OCC	*Occlusion*. Object becomes partially or fully occluded.
OV	*Out-of-view*. Object is partially clipped by the image boundaries.
SC	*Shape Complexity*. The object has complex boundaries such as thin parts and holes.
SV	*Scale Variation*. The area ration among any pair of bounding-boxes enclosing the target object is smaller than 0.5.

Table 3.2 List of video attributes and their descriptions as provided by Perazzi et al. (2016). Each video sequence is annotated with one or more of the attributes present in this table.

Each of the video sequences is annotated with a set of one or more of the attributes, listed in Table 3.1, present in that video. This makes a more meaningful qualitative evaluation of any proposed solution possible.

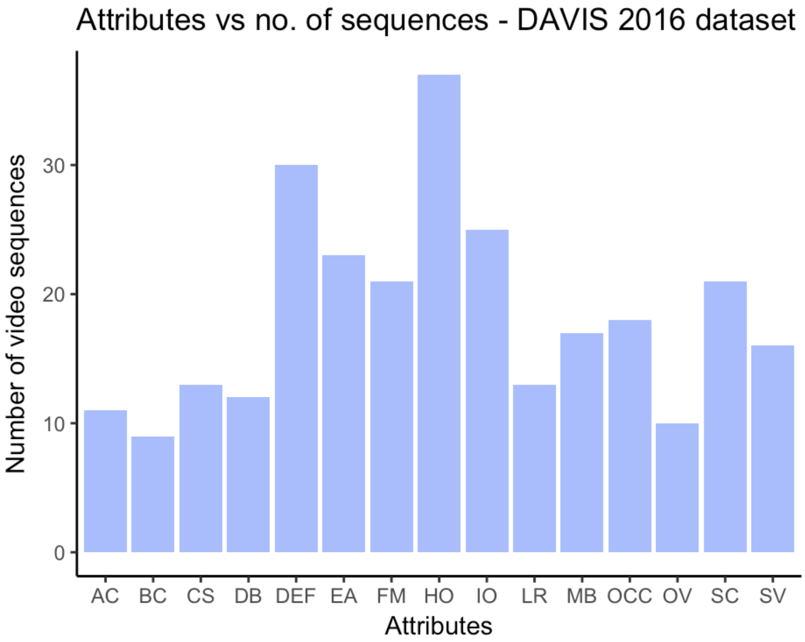

Figure 3.2 Distribution of attributes across sequences - DAVIS 2016 dataset

The video sequences are available in two resolutions – full HD (1920 X 1080) and FWVGA (854 X 480). This research will be conducted with the FWVGA resolution sequences keeping the computational complexity in mind.

3.4. Evaluation Criterion

The metric that will be used to quantitatively evaluate the models built in this research is the mean Jaccard Index over the frames. Jaccard Index is defined as *"intersection-over-union of the estimated segmentation and the ground-truth mask"* (Perazzi et al., 2016). This measure thus gives a good idea of how well the predicted mask fits the ground truth. For each frame, the Jaccard Index is computed by dividing the total number of pixels that are common to both the predicted mask and the ground truth over the number of pixels that fall under either the predicted mask or the ground truth. The mean Jaccard index is computed over multiple frames – it is defined as the ratio of total number of pixels that fall in the intersection of any ground-truth – segmentation pair over the total number of pixels that fall in any ground-truth or segmentation across the frames considered.

3.5. Research Design

The experiment can be divided into three principal phases. The first phase involves the development of an evaluation framework that returns the mean Jaccard index across multiple frames, given the respective sets of ground-truths and segmentations. In the second phase, a solution for the video object segmentation problem is implemented by extending the original Mask RCNN framework to perform video object segmentation in the semi-supervised environment of DAVIS 2016 dataset. The final phase involves the modification of the Mask RCNN framework to use Exponential Linear Units (ELU) instead of the default Rectified Linear Units (ReLU). The results of both the phases are recorded. The experiment is followed up with a comparison of the results obtained in the two phases, and an evaluation (both qualitative and quantitative) of the two techniques. The research is then concluded by summarising the findings.

4. RESULTS, EVALUATION AND DISCUSSION

4.1. Introduction

This chapter discusses the implementation details and the results of the experiments conducted. It starts off by providing a high level explanation of Mask R-CNN, the image instance segmentation framework chosen for the research. It proceeds to explain about the data preparation steps taken in the research – the application of pre-trained weights, video data preprocessing steps and splitting the data into training, validation and testing subsets. This is followed by an explanation of the implementation details such as the software environment used for the research, code structure, and other software development efforts. The chapter concludes with a reporting of the results observed in the experiments – mean Jaccard indices over various groupings of the data, and an evaluation of the results.

4.2. Mask R-CNN

4.2.1 Network Architecture

Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017) is a CNN-based approach that aims to solve the object instance segmentation problem in images. Mask R-CNN architecture can be divided into two – a backbone, and a head. The backbone is a deep convolutional network that specialises in understanding spatial patterns and identifying features. This architecture is modular in that the backbone can be altered if needed, a classical deep convolutional network stripped of the final fully connected layers could act as the backbone. The head consists of the three smaller networks, performing object classification, bounding box regression and mask prediction respectively.

Mask R-CNN directly extends on Faster R-CNN (Ren, He, Girshick, & Sun, 2015). Faster R-CNN attempts to tackle the problem of object detection, Mask R-CNN extends Faster R-CNN by adding a Fully Convolutional Network (Long, Shelhamer, & Darrell, 2014) branch to predict a mask for each instance. A key difference between the Mask R-CNN approach and a completely FCN-based approach towards object segmentation is that the task of instance segmentation is completely decoupled in

Mask R-CNN in that the network predicts a binary mask for each class independently, unlike the FCN-based approach in which there is a pixel-level competition between the classes, thus coupling the classification and segmentation tasks together. Mask R-CNN is based on an instance-first strategy, rather than on segmentation-first strategies.

4.2.2 Faster R-CNN

The Faster R-CNN architecture can be divided into two stages. The first stage is a Region Proposal Network (RPN) tasked with object detection and localisation. This RPN stage works by coming up with the candidate bounding boxes, or the Regions of Interest (RoIs). The second stage of Faster R-CNN is a Fast R-CNN (Girshick, 2015) which aims at classification and bounding box regression. RPN in itself an FCN which shares the full convolutional features with the Fast R-CNN object detection framework. Faster R-CNN has two outputs for each candidate object, a class label and a bounding box offset. Mask R-CNN adds a third FCN branch that outputs the mask of the candidate as well. This is depicted in Figure 4.1.

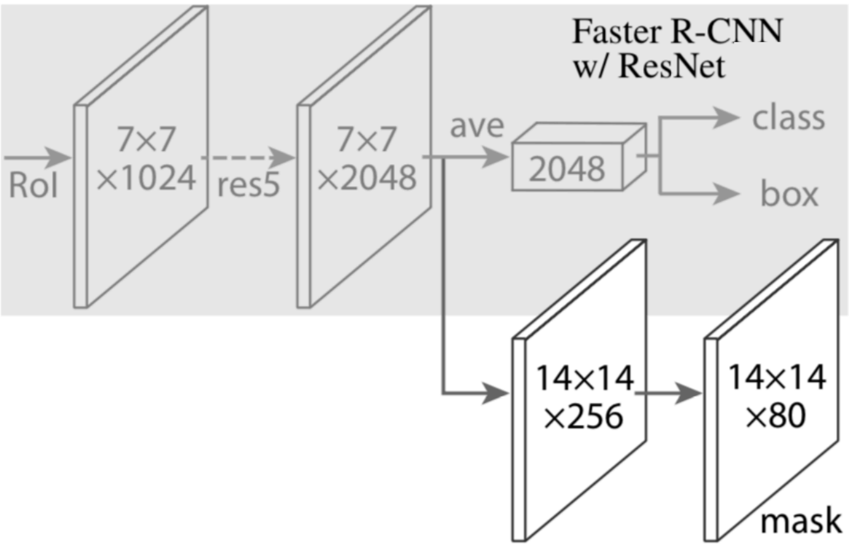

Figure 4.1 The architecture of the Mask R-CNN head used in this research. The image is taken from the original Mask R-CNN paper (He et al., 2017)

Mask R-CNN uses the same first stage as Faster R-CNN, the RPN. The second stage (the Fast R-CNN one) is altered to incorporate the third FCN branch as well. The intuition behind this is that the spatial features learned by the deep FCN backbone can be shared for all the three tasks. The authors discuss two architectures that can be used as the backbone for Mask R-CNN – ResNet (He et al., 2015) and Feature Pyramid Network (Lin et al., 2016). This research uses an implementation with ResNet for the backbone as the authors discuss about the remarkable advantage in both accuracy and speed a ResNet backbone architecture has over its FPN counterpart.

This research goes ahead with the hyperparameters suggested in the original Mask R-CNN architecture, which the authors have based on empirical evidence and extended from the original Fast R-CNN and Faster R-CNN architectures. Though how the tuning of these hyperparameters impact the training would be a fruitful undertaking, it is beyond the scope of this research, and as such, only hyperparameter adjustments deemed absolutely necessary due to memory constraints were made. Any such implementation choice made is discussed further later.

4.2.3 The loss function

The loss L of the entire Mask R-CNN network is defined as the sum of losses over the three constituent networks:

$$L = L_{cls} + L_{box} + L_{mask}$$

The classification and bounding box regression losses. L_{cls} and L_{box} are defined exactly as in the original Fast R-CNN architecture. The mask branch outputs K m × m binary masks for each RoI, one for each of the K classes. The loss over this branch L_{mask} is calculated using the mask predicted for the ground truth class by the branch. A per-pixel sigmoid is applied to this mask and L_{mask} is defined as the average binary cross-entropy loss. This is a key difference between the Mask R-CNN approach to the problem and the more traditional FCN-based solution to segmentation. The FCN-based solution (Long et al., 2014) employs a per-pixel softmax classifier and the loss function used is a multinomial cross-entropy loss – this creates a competition for a pixel between the object classes. In the Mask R-CNN approach, this competition is avoided.

4.2.4 RoIAlign

The FCN branch used for the mask prediction requires the smaller feature maps (RoI features) extracted to be well-aligned to preserve the explicit per-pixel spatial correspondence. This is achieved with the help of RoIAlign, a novel approach introduced along with the Mask R-CNN architecture. RoIPool, which is the technique used in Faster R-CNN, extracts a small feature map from an RoI. But, the continuous discretisation applied by RoIPool can negatively impact the task of pixel-perfect mask prediction. RoIAlign solves this problem by removing the harsh quantisation performed in RoIPool.

4.3. One-shot fine-tuning

Caelles et al. (2016) introduced an interesting approach to the problem of semi-supervised object segmentation in videos. The modular approach could be divided into three phases – selection of a base deep convolutional FCN pre-trained on a large dataset such as the MS-COCO (Lin et al., 2014) or ImageNet (Russakovsky et al., 2014) datasets, training the entire network on the training dataset available for the specific object segmentation task, and finally fine-tuning the entire network on the available frames and ground-truth pairs for the test dataset. The DAVIS 2016 semi-supervised video object segmentation problem statement allows the use of the first frame of a validation video sequence. Caelles et al. (2016) also proposed their solution on the DAVIS 2016 dataset itself. The intuition behind this approach of a final fine-tuning of the entire network on the first frame is to attune the network to focus on the principal object in the videos.

This research implemented a similar solution, in which the base network is the Mask R-CNN network (ReLU and ELU models were tested) pre-trained on the MS-COCO dataset. The second step of training on the specific dataset (DAVIS 2016) was carried out, initially training the head of the network till the validation error was minimised, and then the entire network (backbone + head) was fine-tuned on a single epoch over the entire training dataset. The final step, during testing, involved a dynamic (one-shot) fine-tuning of copies of the model from the second step on the first frame of the test video sequence and its ground-truth, before predicting the segmentation.

4.4. Data preparation

4.4.1 Pre-trained weights

Deep Learning has been applied to a host of domains to solve a wide range of problems to increasing degrees of success. As mentioned before, the training of models is a possibly time-consuming endeavour and many a times, doing so from scratch is avoided if it can be. This is made possible by the different pre-trained models and/or weights made available by the respective researchers. These pre-trained weights could be used as benchmarks and allow testing a newly built model to be compared against the baseline. Also, a model that has been trained on a particular dataset to solve a specific problem could have learned some features that are dataset-independent and problem-specific. Thus, the weights learned during the training on a dataset could be proven useful if used properly to solve the same problem (or even a similar one) on another dataset. The use of pre-trained weights could help in saving time during training, providing a starting point in the right direction instead of starting from zero.

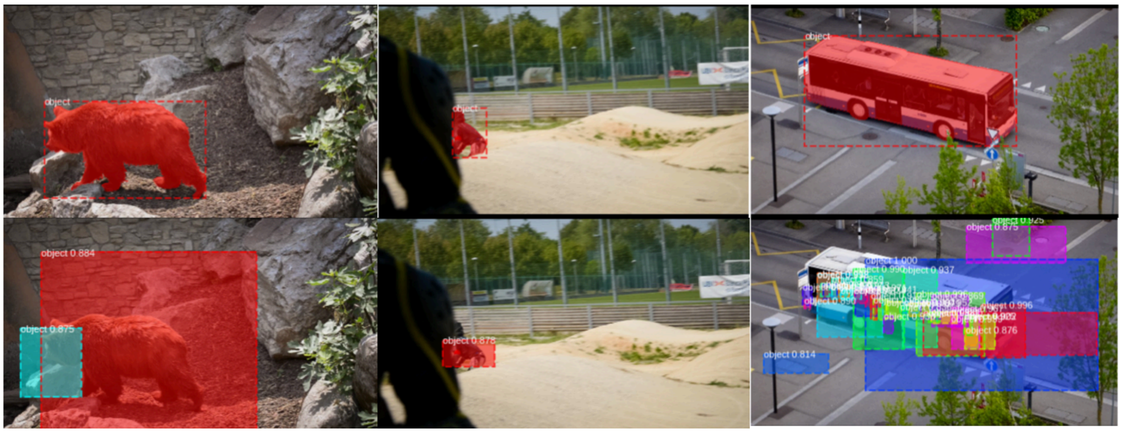

Figure 4.2 Some sample segmentations before training on the DAVIS dataset — the top row shows the images superimposed with the respective ground truths, the bottom row shows the images superimposed with the predicted segmentations of identified objects by the model with only the pre-trained backbone weights.

The Microsoft Common Objects in Context dataset (Lin et al., 2014), shortened to MS COCO, is a dataset that has around 330,000 images, of which over 200,000 are labelled, with around 1.5 million object instances spread across 80 object categories. A backbone of ResNet-101 (He, Zhang, Ren, & Sun, 2015) was used for the Mask R-CNN model in this research. Weights for the specific backbone used in this implementation (Waleed Abdulla, 2017) pre-trained on the MS COCO dataset has been made available for use. These weights have been used in this research to aid with the training phase. The total number of pre-trained backbone weights loaded across the network 63,733,406. Figure 4.2 shows the performance of the Mask R-CNN model on some images from the training dataset with only the pre-trained backbone weights.

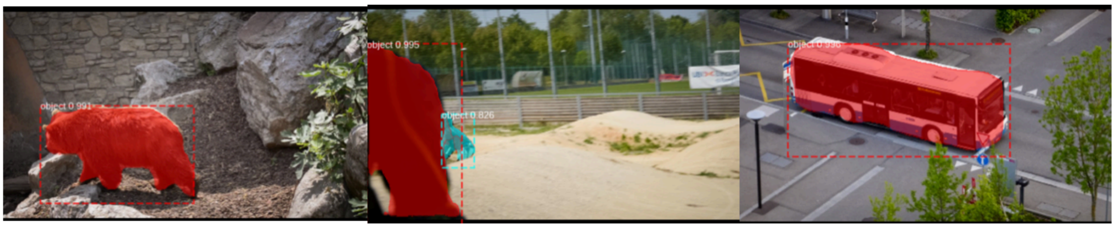

Figure 4.3 Segmentations for the same images after one additional epoch of training the head of the network architecture on the DAVIS dataset.

As can be seen, the model already performs quite well in locating the object with just the pre-trained weights and no training. Figure 4.3 shows the performance of the same model on the same images after one additional training epoch for just the head of the network on the DAVIS 2016 dataset. Thus, this has sped up the training process by a considerable amount.

The weights pre-trained on MS-COCO dataset available were for the Mask R-CNN architecture with the ReLU activation units. For the second model based on ELU, a similar Mask R-CNN model instance with ELU activation units, and randomly initialised weights was trained on the MS-COCO dataset. Post this training, the weights for the backbone were preserved and the head were re-initialised with random weights, to make the training process similar to the one for the model with ReLU activation. The rest of the training was exactly the same as that of the ReLU-based model.

4.4.2 Data preprocessing

4.4.2.1 Frame transformation for conforming to Mask R-CNN requirements

Data preprocessing is an integral step in any data mining or machine learning task that aims to make the raw data acquired (either primary or secondary) to a format that is required by the application for consumption. This is required because the data available in the real world may not only be unfit for usage, but also partial or unclean. Specifically in the domain of video object segmentation, data preprocessing would primarily involve taking in the video(s) and converting it into a data format that can be understood by the object segmentation model. For instance, there could be video sequences of varying frame resolution present in a dataset, thus possibly making a preprocessing step required in which this is handled and all the video sequences converted to a consistent frame resolution before being passed to the model.

The DAVIS 2016 dataset consists of 50 video sequences, all captured originally at 24 frames per second at a resolution of 480p. The dataset structures the data so that all the frames are contained in a folder and are named numerically, in ascending fashion. Each frame is presented as a JPEG image.

The OpenCV (Open Source Computer Vision Library) Python interface is used to read the frames into the application. A frame is read in as a numpy array of 1229760 values and specifically of shape (480, 854, 3). The shape is of (height, width, channels) format. 480 and 854 represent the resolution of the 480p frame – that is the total number of pixels present in the frame. The 3 indicates the three channels of a colour image – red, green and blue. Each element in this array could take a value from 0 to 255, indicating the intensity of that specific channel in its pixel. For example, a red pixel would have (255, 0, 0) as the RGB channel values associated with it.

The primary image preprocessing step performed was to transform the frames into a shape that was required by the Mask R-CNN implementation that was used for the research. The implementation required a square shaped frame with each dimension a multiple of 2^6. The closest dimension that met these requirements were 832 X 832 pixels. Figure 4.4 shows this preprocessing step. Thus, the shape of the resized frame is (832, 832, 3). This is achieved by padding an equal number of 0s at the top and bottom of the original image to make it 832 pixels. The original height of the image being 480 pixels, (832 - 480) / 2 = 176 more 0 pixels (3 channels of all zeros) are added to the top and the bottom of the frame.

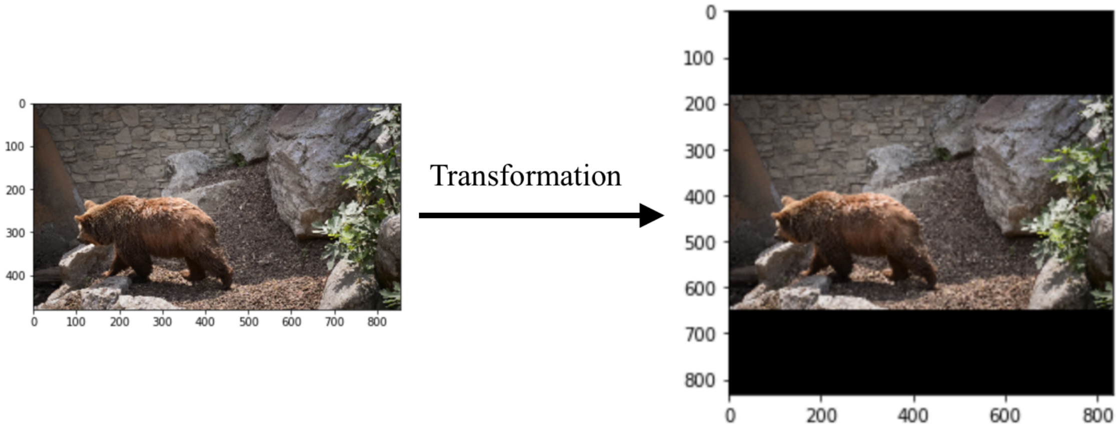

Figure 4.4 Image transformation performed to conform to the requirements of the Mask R-CNN model. The rectangular image is converted to a square shaped one by padding zeros at the top and bottom.

4.4.2.2 Ground-truth and segmentation transformation for evaluation

The ground-truths provided by DAVIS 2016 are of the same shape as that of the frames, (854, 480, 3). The same image transformation process, explained in the previous section, is carried out for the ground-truths as well, converting them to (832, 832, 3).

Also, the segmentations predicted by Mask R-CNN are of shape (832, 832, 1). This was reshaped to (832, 832, 3) by broadcasting the first two dimensions of the array to the third.

4.4.3 Splitting the data into training, validation and testing subsets

The DAVIS 2016 dataset, as discussed in section 3.3, consists a total of 50 video sequences. Perazzi et al. (2016) suggests a split of the dataset into a training –

validation split of 30 – 20. Figure 4.5 shows the way the different annotated attributes (challenges posed) are spread across the training and validation splits.

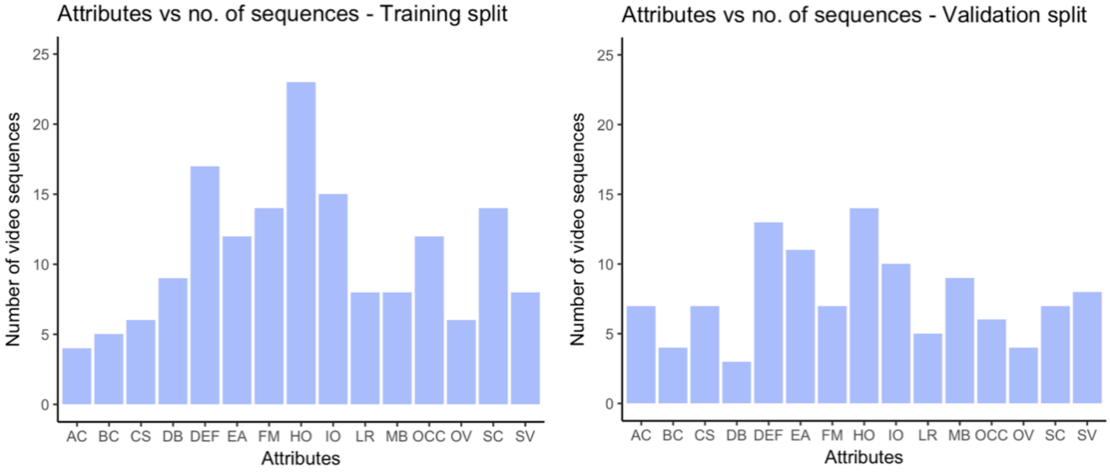

Figure 4.5 Distribution of attributes over training and validation datasets.

The initial method the dataset was split into training, validation and test datasets was by keeping the training data as it is and then randomly splitting the validation data into validation and test subsets in a 3:1 ratio, 15 video sequences in the validation subset and 5 in the test subset. The motivation for this split was to make sure that the robustness of any trained model is validated on a large enough validation subset. The attributes distribution over the test subset after this split is shown in Figure 4.6. As can be seen, three of the fifteen attributes are not represented in the test subset (BC, CS, DB), and another three attributes are seen in only one video sequence each (LR, OCC, OV). This is not desirable, as this hinders a more meaningful qualitative evaluation of the model.

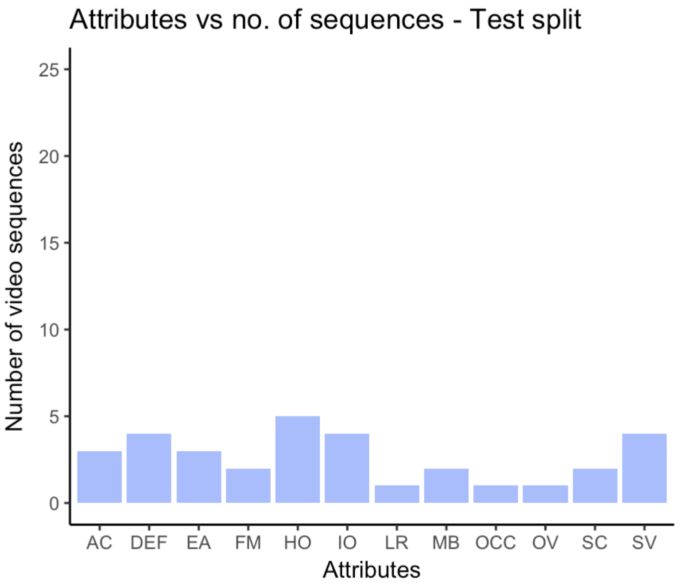

Figure 4.6 Distribution of attributes over test dataset after the initial split.

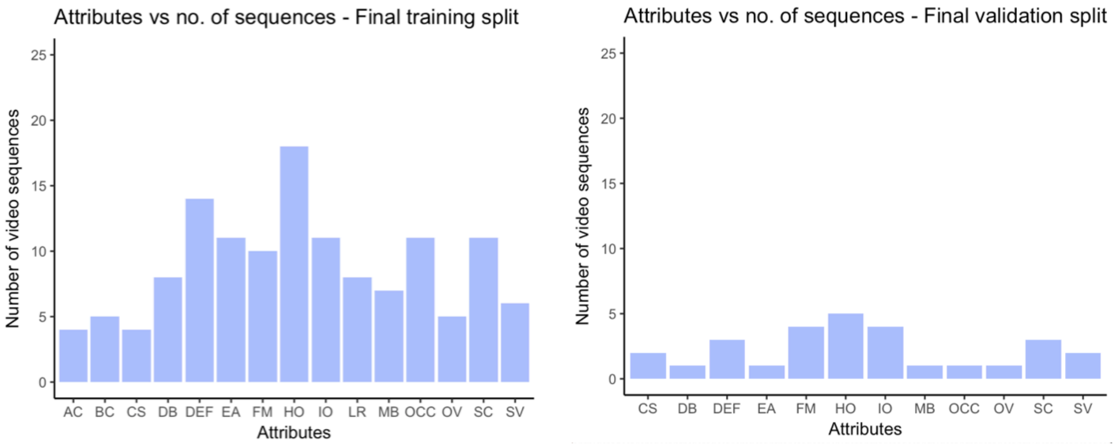

Figure 4.7 Distribution of attributes over training and validation dataset after the final split.

To avoid certain attributes being dropped so that a more meaningful qualitative evaluation can be conducted, the suggested validation dataset of 20 was considered as the test dataset and the training dataset was split into training and validation subsets of 25 and 5 video sequences respectively. Reduction of the training dataset size might work towards making the model more generalised to previously unseen data. Final distribution of the attributes in the training and validation subsets are shown in Figure

4.7 respectively. The distribution in the testing dataset would be the same as the original validation dataset, that is shown in Figure 4.5.

4.5 Mask R-CNN Implementation

4.5.1 Software environment

The primary programming language used for this research was Python 3.6.3. Python was adopted due to its standing as one of the most used programming languages in the domains of machine learning in general and Deep Learning and Computer Vision in particular. Another reason for choosing Python was the wide range of third-party libraries available that facilitates the entire development process.

An implementation of Mask R-CNN by Matterport Engineering team (Waleed Abdulla, 2017) was used. This implementation was written in Python 3.x and is widely considered as one of the best implementations of Mask R-CNN by the Computer Vision development community. The deep neural networks are implemented using Keras, an open source Deep Learning library written in Python. Keras is currently compatible with Python versions up to 3.6. This was another incentive for choosing Python 3.6.3 for the development. Keras can be thought of as a higher level interface rather than a lower level library in that it runs on top of other core Deep Learning libraries like TensorFlow, Microsoft Cognitive Toolkit, or Theano. In this implementation, the Keras interfaces with Tensorflow, an open source machine learning framework developed by the Google Brain team (Abadi et al., 2016), keeping the necessity for high speed parallel computations that facilitate quick performing of vectorised operations which form the crux of the various Deep Learning operations. Specifically, Keras 2.2.4 and Tensorflow 1.12.0 were used for the development, both the latest stable versions of the respective libraries at the time of writing.

Jupyter notebooks were used heavily for exploring and working out different aspects of the data and understanding the original Mask R-CNN implementation. Jupyter notebook is an open-source web-based tool provided by Project Jupyter that builds on the powerful Interactive Python (iPython) shell and facilitates the combining of code, outputs and any narrative or explanatory text. Normal iPython shell was used for similar purposes due to the author's prior familiarity with it. An advantage of using Jupyter notebooks is the ease of sharing a notebook between developers and also persisting the interactions with the compiler, thus enabling the work to be picked up from where it was left off at a later point of time comfortably. Sublime Text 3, a simple yet powerful text editor was also used to write Python scripts during the development.

Training a deep neural network is a computationally intensive task and involves a high amount of operations. Due to vectorisation, these operations could be done concurrently. Although, the sheer magnitude of the number of operations required, even if simple enough the individual operations are, makes an extremely large amount of memory required. For instance, one forward pass of the Mask R-CNN network with a ResNet 101 backbone and its head involves learning more than 63 million parameters for a single image (a mini-batch size of 1). A Graphics Processing Unit (GPU) has proven to suit the Deep Learning operations much better than the normal Central Processing Unit (CPU). Although the GPU core need not be faster than the CPU core, the larger number of cores present in the GPU and the faster memory makes the execution of parallel operations faster in a GPU (sequential code would be executed faster in a CPU than in a GPU). Thus, a GPU environment was required to train the deep network. Colaboratory, a Google research project aimed to aid in machine learning education and research, was used as the primary environment to carry out the various experiments. Colaboratory provides its users with Tesla K-80 GPU with GPU Random Access Memory (RAM) of 12GB.

R, an open source software environment for statistical computing and graphics, was used to visualise various aspects of the data during the research. The visualisations presented in this dissertation are also produced using R. R 3.4.2 running on RStudio 1.0.153, the most popular Interactive Development Environment (IDE) for R, was used for all the visualisations. This choice was made due to the author's comfort with ggplot2 (Wickham, 2016), a popular data visualisation package in R.

Git, a popular version control software, was used in collaboration with GitHub, to maintain and manage the different versions of code during the development, and to easily make it available in the Colaboratory environment.

4.5.2 Code structure

The implementation is structured so that it spans across three primary modules. These are –

1. utils – all the utility functions that help in the smaller specific tasks are written here.
2. config – the various hyperparameters involved in the training goes here. The different hyperparameters available that were relevant to the research are listed and described below. Most of the hyperparameters were left untouched as they were already tested out and set according to prior experimental evidence.
   a. NAME - a custom name that can be given to identify a particular experiment, useful when multiple experiments are run.
   b. GPU_COUNT - the number of GPUs to use.
   c. IMAGES_PER_GPU - the number of images to train on each GPU instance. The mini-batch size for a pass would be GPU_COUNT * IMAGES_PER_GPU
   d. STEPS_PER_EPOCH - the number of iterations when it should be considered as an epoch. This need not be set to the size of the training dataset – this has been written thus so that validation steps can be made in a higher frequency if needed.
   e. BACKBONE - resnet50 and resnet101 are the supported values. This research uses resnet101.
   f. NUM_CLASSES - the number of object classes used in training plus the background class. This has been set to 2 for this research, an object class and a background class.
   g. USE_MINI_MASK - whether or not to scale down the instance masks to a smaller size to reduce the load on memory. This is recommended when using images of higher resolution. This was set to True for this research.

h. MINI_MASK_SHAPE - dimensions of the resized mask – the default value is 56 X 56 and has been left unchanged in this research.

i. IMAGE_MAX_DIM - the upper limit to which the image is padded up with zeros to make it square shaped and a multiple of 64 (as discussed in section 4.4.2.1)

j. IMAGE_CHANNEL_COUNT - the number of channels in a frame in the dataset. This has been set to 3 as each frame in the DAVIS 2016 dataset has three channels (RGB).

k. DETECTION_MIN_CONFIDENCE - the percentage confidence threshold required for a region of interest to be presented as a detected instance.

l. LEARNING_RATE - learning rate of the algorithm. Set to 0.001 after trying out different values among [0.0001, 0.0003, 0.001, 0.003, 0.01], keeping the rest of the hyperparameters unchanged. 0.001 was found to be a good learning rate because the learning was not too slow and it did not fail to converge. The Mask R-CNN paper uses a learning rate of 0.02, but it is advised against by Waleed Abdulla (2017) in this implementation as it causes the weights to explode possibly due to different optimiser implementations.

m. LEARNING_MOMENTUM - the value of momentum to be used in the learning. The default value of 0.9 is used.

n. WEIGHT_DECAY - the value of weight decay to be used to prevent explosion of weights. The default value of 0.0001 is used.

o. GRADIENT_CLIP_NORM - the threshold value used to clip the gradient, to prevent the abnormal growth of the gradients. The default value of 5.0 is used.

3. model – the Mask R-CNN network architecture is written here.

4.6 Results

The Mask R-CNN model with ReLU activation units recorded an average Jaccard Index of 0.709 while that with ELU activation units recorded an average Jaccard Index of 0.707.

Figures 4.8 and 4.9 show the mean Jaccard indices observed over each video sequence in the testing dataset by the model with ReLU activation units and the one with ELU activation units respectively. Table 4.1 compares the mean Jaccard indices observed over each video sequence for further clarity.

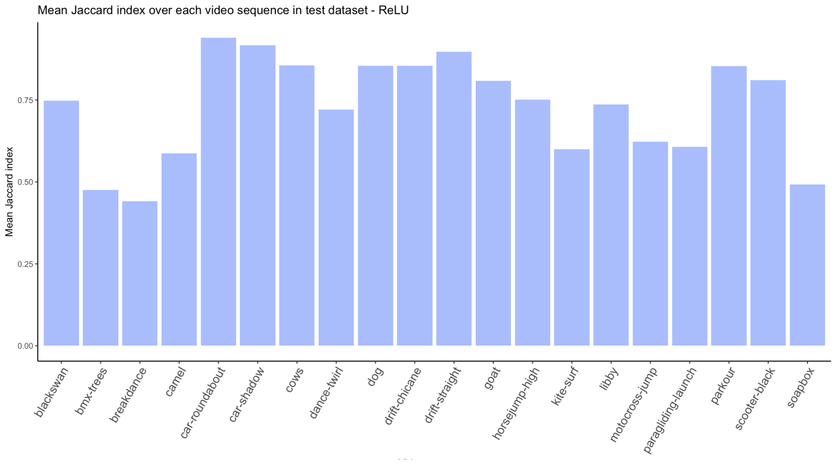

Figure 4.8 Mean Jaccard indices over test video sequences for model using ReLU as the activation unit.

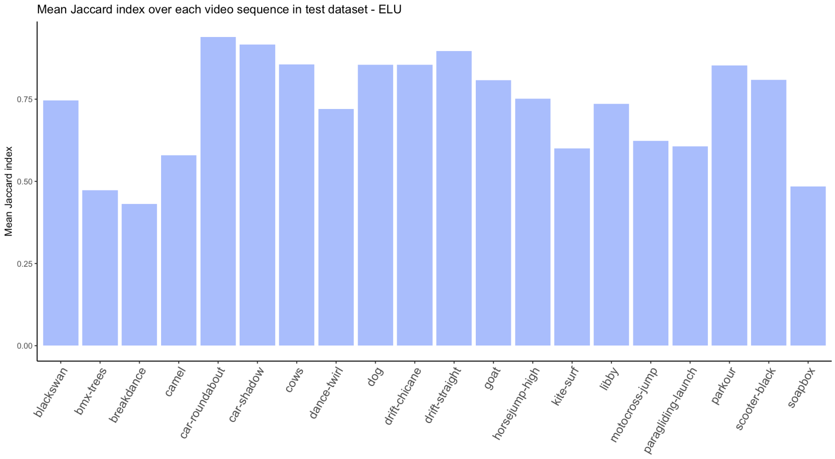

Figure 4.9 Mean Jaccard indices over test video sequences for model using ELU as the activation unit.

Test video sequence	Mean Jaccard index with ELU	Mean Jaccard index with ReLU
blackswan	0.7460477	0.7478864
bmx-trees	0.4731706	0.4754088
breakdance	0.4310772	0.4409379
camel	0.5790564	0.5871835
car-roundabout	0.9395416	0.9403327
car-shadow	0.9162216	0.9165938
cows	0.8559922	0.8556838
dance-twirl	0.7198102	0.7203358
dog	0.8551628	0.8545649
drift-chicane	0.8546576	0.8544595
drift-straight	0.8961719	0.8974399
goat	0.8079687	0.8080276
horsejump-high	0.7515617	0.7516226
kite-surf	0.6004799	0.5993657
libby	0.736075	0.7366984
motocross-jump	0.6233499	0.6230072
paragliding-launch	0.6064516	0.6068224
parkour	0.8522469	0.8528893
scooter-black	0.8092873	0.8102524
soapbox	0.4846367	0.4925435

Table 4.1 Mean Jaccard indices over test video sequences for models using ReLU and ELU as activation units — a comparison.

4.7 Analysis, Evaluation & Discussion

From observing the Figures 4.8, 4.9 and Table 4.1, it is evident how closely the two models compare in their respective performances. Both the models scored strongly across same video sequences. So, it would suffice to conduct an error analysis on one of the models. This section looks at what the observed results mean in the context of the research question and the null hypothesis, and then proceeds to try and understand more about the working of the model that used ReLU activation units, and identify the strengths and weaknesses of the model.

4.7.1 Quantitative analysis

Both the models posted overall very similar mean Jaccard index over the test dataset – the model with the ELU activation units posted 0.707, where the one with the ReLU activation units posted 0.709. From these observed results, **the research has failed to reject the null hypothesis that stated that there would be no significant impact in the observed mean Jaccard index if the more traditional ReLU activation units are replaced with ELU activation units in a deep convolutional architecture tasked with semi-supervised object segmentation in videos.**

4.7.2 Error analysis

The model posted a mean Jaccard Index of 0.709 over the test dataset, meaning 70.9% of the pixels classified as the prominent object across the 1376 frames fell in the intersection of the respective ground-truth - segmentation pairings.

4.7.2.1 What went well?

The two video sequences that the model performed the best both featured cars, car-roundabout (0.940) and car-shadow (0.917) were the sequences that the model recorded the best scores. Figure 4.10 and 4.11 shows how the predictions for these sequences compare with the respective ground-truths.

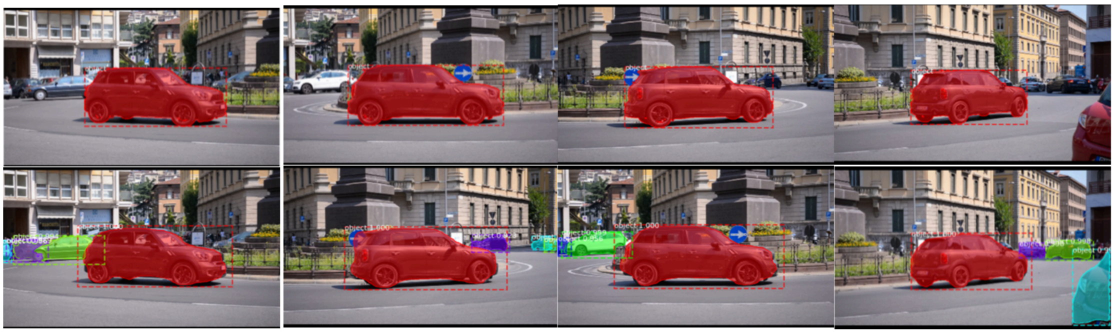

Figure 4.10 Ground-truth vs predicted segmentations for car-roundabout video sequence. The top row shows frames from the sequence at an interval of 20, superimposed with the ground-truth and the bottom row shows the same frames superimposed with the predicted segmentations.

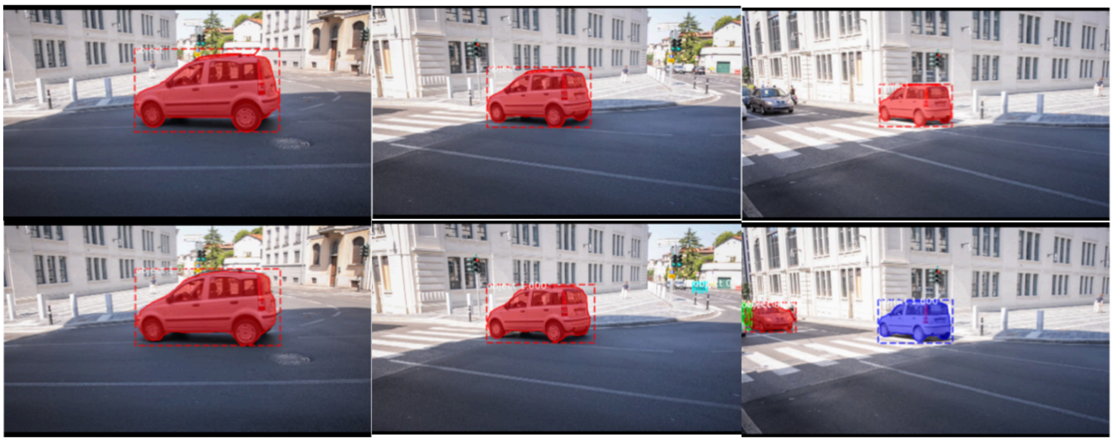

Figure 4.11 Ground-truth vs predicted segmentations for car-shadow video sequence. The top row shows frames from the sequence at an interval of 20, superimposed with the ground-truth and the bottom row shows the same frames superimposed with the predicted segmentations.

In both these video segments, it can be observed that the principal object to be detected are centred in the frame and the objects (cars) are conveniently placed in the forefront as well. Other than some background clutter, there is not much of a challenge in these video sequences. On examining the extra annotations provided in the dataset, the car-roundabout sequence is marked as indeed having BC (Background Clutter), and the car-shadow sequence is marked as afflicted by four challenges (Appearance Change, Background Clutter, Edge Ambiguity, Low Resolution). Even though the model performed extremely well in segmenting the principal objects in these videos, it is interesting to observe that some other objects in the background are also identified and segmented by the object. This did not hurt the score in this case because the principal object is still the object with the highest probability score (confidence) assigned to it by the model and thus, the other segmentations are discarded. This could have been a problem if there was an object segmented by the model that had a higher confidence assigned to it – the mask of that object would have been the one used to compute the Jaccard Index.

4.7.2.2 What went wrong?

Next, the analysis looks at two of the video sequences for which the model performed the worst. These are bmx-trees (0.475) and breakdance (0.441). Figure 4.12 and 4.14 shows how the predictions for these sequences compare with the respective ground-truths.

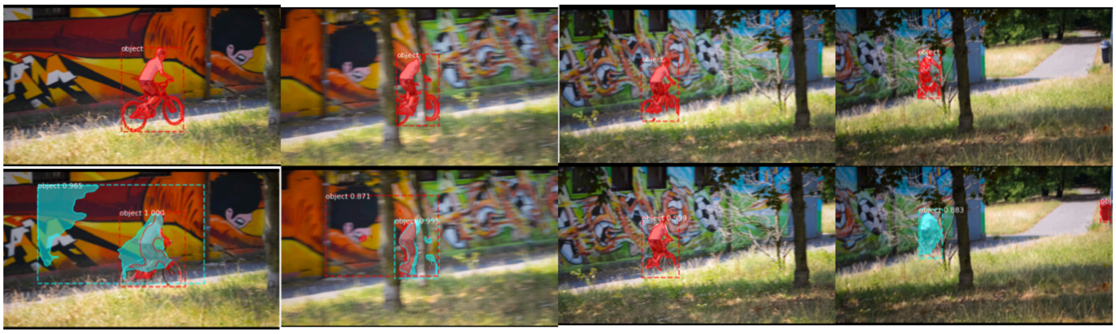

Figure 4.12 Ground-truth vs predicted segmentations for bmx-tree video sequence. The top row shows frames from the sequence at an interval of 20, superimposed with the ground-truth and the bottom row shows the same frames superimposed with the predicted segmentations.

The bmx-tree video sequence is attributed with 12 of the possible 15 attributes, the maximum for any video sequence in the dataset, making it one of the most complex video sequences the model has come across. From figure 4.12, some of the challenges in the video sequence are evident, like the background clutter presented by the graffiti-covered wall and the occlusion of the bike and the rider by the tree. Still, it appears that the model has performed relatively very well on the final two frames presented in the figure.

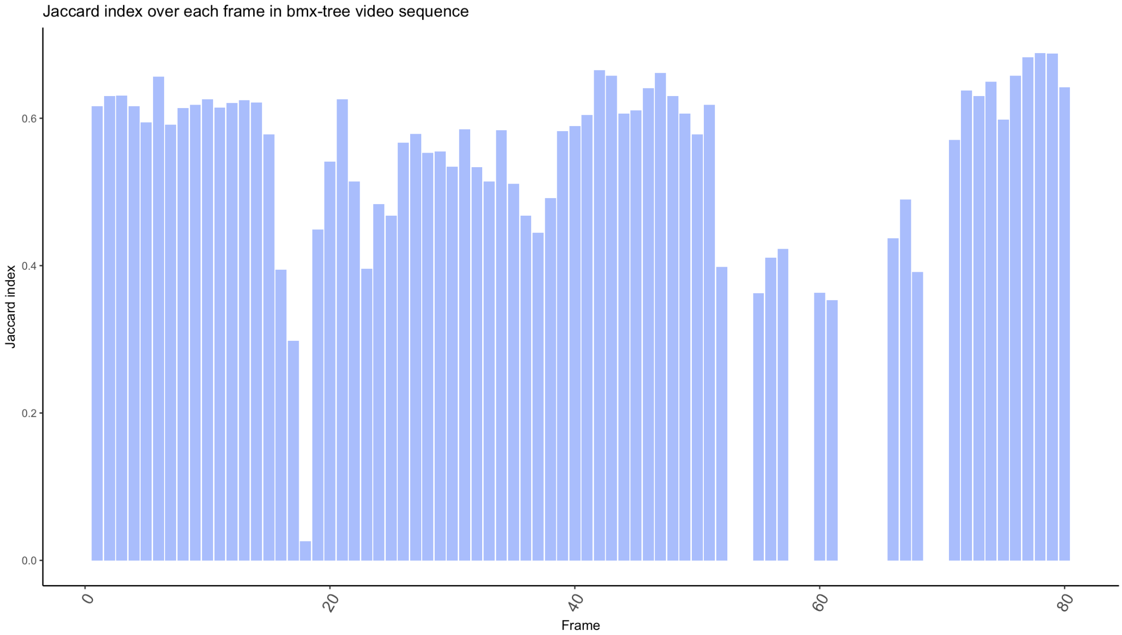

Figure 4.13 Jaccard index over each frame in the bmx-tree video sequence.

Figure 4.13 shows the Jaccard index observed across each frame in the bmx-tree video sequence. As can be seen, the performance drops tremendously from frame 50 to frame 70. This is the region where the object (bike and rider) is occluded by the tree. The model performs moderately well through the rest of the sequence, even when heavy background clutter is encountered. Thus, in this video sequence, occlusion seems to be the most difficult challenge for the model.

From figure 4.14, it seems that even though the model did correctly identify the principal object in the video, the performance was hurt by the same phenomenon that was observed for the car-roundabout sequence as well. The model identified and segmented objects that are not principal to the video. That this segmentation of non-principal objects happened even after the fine-tuning on the first frame and its ground truth is definitely a limitation of the current model though and needs to be looked into.

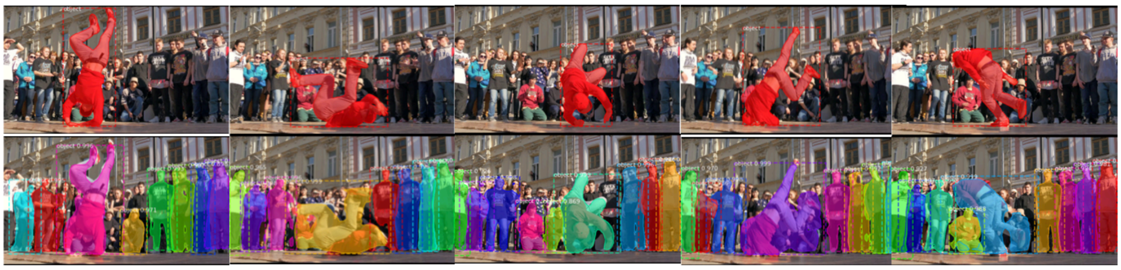

Figure 4.14 Ground-truth vs predicted segmentations for breakdance video sequence. The top row shows frames from the sequence at an interval of 20, superimposed with the ground-truth and the bottom row shows the same frames superimposed with the predicted segmentations.

4.7.2.3 Does video duration affect the mean Jaccard index?

Figure 4.15 shows the variation of mean Jaccard index with variation in duration of the video sequence. At a quick glance, there does not seem to be any correlation between the two.

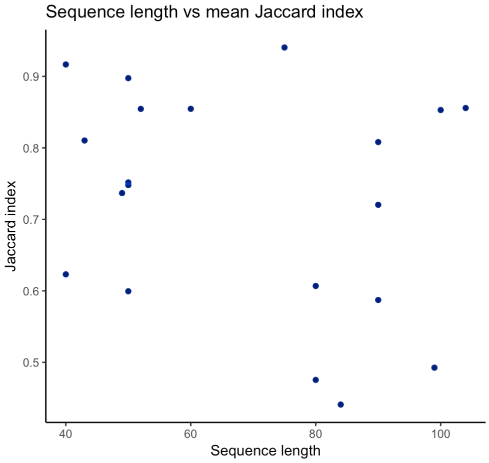

Figure 4.15 Scatterplot visualising the variation of mean Jaccard index over sequences with their lengths.

No assumptions were made about the normality of the data and a Kendall's rank correlation test was conducted to examine the relationship between length of a video sequence and its mean Jaccard index. The statistical significance considered was 0.05 for the test (Field, Miles and Field, 2012). A moderate negative association (Cohen, 1988) was observed between the length of a video sequence and the mean Jaccard index reported for it, however the result was not statistically significant, τ_b = -0.11, p = .49)

4.7.2.4 Attributes and mean Jaccard index - any obvious patterns?

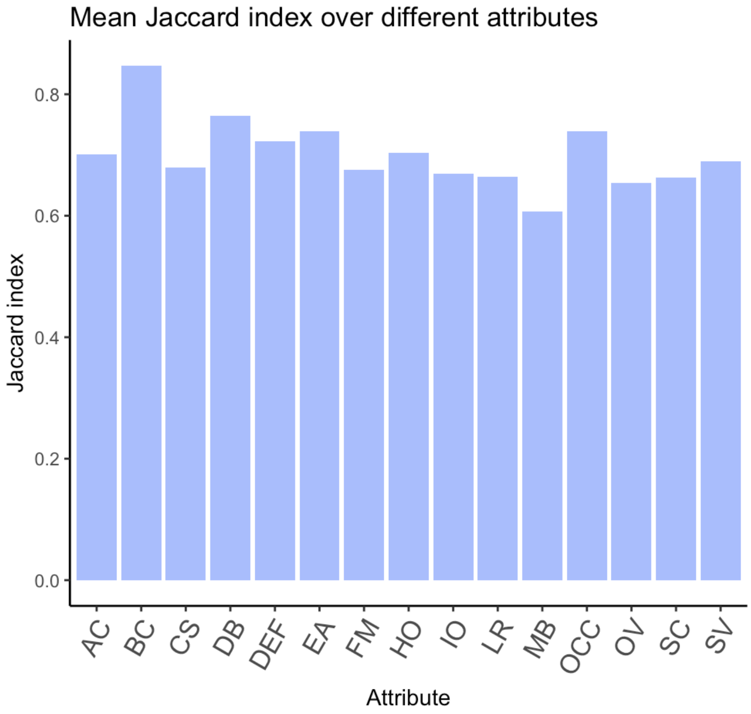

Figure 4.16 Mean Jaccard index aggregated over videos grouped under the same attributes.

Figure 4.16 shows the mean Jaccard index over the fifteen different annotated attributes in the dataset. No clear pattern seems to emerge from the visualisation.

5. CONCLUSIONS

5.1 Research Overview

This research aimed at understanding the problem of video object segmentation in general, and in a semi-supervised environment in specific. The primary goal of the research was to gain a deep level of understanding on how an image object segmentation algorithm based on deep learning techniques could be extended to solve the same problem in videos, and to investigate the impact of changing the activation function from Rectified Linear Units (ReLUs) to Exponential Linear Units (ELUs) on the performance of a model.

The research started off with a comprehensive review of the relevant literature, which in itself was a great learning process, followed by the identification of a proper dataset. DAVIS 2016 dataset was chosen because it is widely identified as the benchmark in object segmentation currently. Due to the author's prior experience in Python and Python's popularity in the Deep Learning and Computer Vision communities, Python 3.6 was identified as the programming language to proceed with. Rest of the development environment was set up based on this initial choice of Python. A popular image segmentation algorithm was identified in Mask R-CNN.

Three phases were identified in the experiment. The first was the deciding on an evaluation metric and designing a framework to compute the evaluation metric given the predicted segmentations and the ground-truths. The evaluation metric decided on to evaluate the models developed was mean Jaccard index.

The second phase was to develop the model with the ReLU activation unit. An implementation of Mask R-CNN by the Matterport engineering team (Waleed Abdulla, 2017) was used as a base for the model being developed. Weights for the backbone of the network pre-trained on the MS-COCO dataset were loaded to speed up the training of the model. Development efforts included understanding the code written and how to extend it to the DAVIS 2016 dataset. This facilitated the training and fine-tuning of the model on the DAVIS 2016 dataset. Training involved just updating the weights of the network head on the DAVIS 2016 training dataset, followed by fine-tuning the whole network with one epoch over the entire training dataset. Once this was done, the next most important step was the development of the one-shot fine-tuning system as implemented in OSVOS (Caelles et al., 2016). This involved dynamically training a new model that has been fine-tuned specifically on the first frame of the test video sequence, in an attempt to zero in on the object of principal focus.

The third phase was to develop a similar model, with just the activation unit changed from ReLU to ELU. The first step in this phase involved the training of the entire network on the MS-COCO dataset for consistency. The weights after this training were retained for the backbone of the network. The rest of the training procedure was the same for this model as it was for the previous model.

Both the models were then used to predict on the video sequences in the test dataset.

5.2 Problem Definition

The research question this thesis set out to investigate was: Can changing the activation unit of a convolutional neural network trained to perform semi-supervised object segmentation in videos from Rectified Linear Unit to Exponential Linear Unit impact the mean Jaccard Index observed for the model?

The null hypothesis formulated to answer the research question was that there is no impact in the mean Jaccard index observed for a convolutional neural network to perform semi-supervised object segmentation in videos when the activation unit of the network is changed from Rectified Linear Unit to Exponential Linear Unit.

The evaluation metric chosen, mean Jaccard index, was recorded for both the models developed in the research over the video sequences on the test dataset. A comparison of these observed values would help to decide on whether or not the null hypothesis can be rejected.

5.3 Results

Both the models developed as part of this research were used to predict on the test dataset and the respective observed mean Jaccard indices recorded. The model with ReLU activation units recorded a mean Jaccard index of 0.709 over the test data whereas the second model with ELU activation units recorded a mean Jaccard index of 0.707. The scores of both the models are comparable and as such, this research has failed to reject the null hypothesis.

5.4 Contributions

The primary contribution of this work is the showcasing of the usefulness of employing Exponential Linear Units in the context of video object segmentation. This is an area that can be further explored and investigated, given that ELU is proven to learn faster than ReLUs and give better results in the context of image classification (Clevert et al., 2015). This could lead to improvement on the current state-of-the-art not only in accuracy, but in speed also.

The secondary contribution of this research is the code base developed as part of this research. The code is somewhat unstructured at the time of writing this thesis, but the author plans to clean it up and make it available with concise documentation and explanation on its working in the recent future. Also, the author hopes the literature review conducted as part of this research could help researchers in the future in jump-starting their understanding of the different approaches that exist to the problem of video object segmentation, and their evolution over time.

5.5 Future Work & recommendations

Even though the developed models were fine-tuned on the first frame of the test video sequence, the models continued to identify the non-principal objects present in the video as well. This impacts the calculation of the evaluation metric of choice, mean Jaccard index.

While this research answers the question it set out to investigate, the conclusions drawn could be corroborated further in the future with the conduction of similar experiments in other datasets too. Experiments could be conducted over some other popular datasets in the domain as well to discover patterns that help understand the performance of the models further.

Modelling of the temporal structure of the videos using a recurrent component could not be developed due to the complexity of the problem and the time constraints. The author could not go past an initial survey of the literature, and understanding the theory behind the said architecture. An experiment in this direction is planned in the recent future, where the primary objective would be to incorporate a recurrent component into the models developed in this research and examine how it impacts the performance.

6. BIBLIOGRAPHY

Babenko, B., Yang, M., & Belongie, S. (2011). Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(8), 1619–1632. https://doi.org/10.1109/TPAMI.2010.226

Badrinarayanan, V., Budvytis, I., & Cipolla, R. (2013). Semi-Supervised Video Segmentation Using Tree Structured Graphical Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, *35*(11), 2751–2764. https://doi.org/10.1109/TPAMI.2013.54

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. Retrieved from https://arxiv.org/abs/1511.00561v3

Bai, X., & Sapiro, G. (2007). A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In *2007 IEEE 11th International Conference on Computer Vision* (pp. 1–8). https://doi.org/10.1109/ICCV.2007.4408931

Ballard, D. H., & Brown, C. M. (1982). *Computer Vision* (1st ed.). Prentice Hall Professional Technical Reference.

Bengio, Y., Courville, A., & Vincent, P. (2012). Representation Learning: A Review and New Perspectives. *ArXiv:1206.5538 [Cs]*. Retrieved from http://arxiv.org/abs/1206.5538

Bouwmans, T. (2009). Subspace Learning for Background Modeling: A Survey. *Recent Patent On Computer Science*, *2*(3), 223–234.

Bouwmans, T., Baf, F. E., & Vachon, B. (2010). Statistical Background Modeling for Foreground Detection: A Survey, 181–199.

Bouwmans, T., El Baf, F., & Vachon, B. (2008). Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. *Recent Patents on Computer Science*, *1*(3), 219–237.

Brox, T., & Malik, J. (2010). Object Segmentation by Long Term Analysis of Point Trajectories. In *Proceedings of the 11th European Conference on Computer Vision: Part V* (pp. 282–295). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1888150.1888173

Brunetti, A., Buongiorno, D., Trotta, G. F., & Bevilacqua, V. (2018). Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, *300*, 17–33. https://doi.org/10.1016/j.neucom.2018.01.092

Brutzer, S., Hoferlin, B., & Heidemann, G. (2011). Evaluation of Background Subtraction Techniques for Video Surveillance. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1937–1944). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/CVPR.2011.5995508

Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., & Van Gool, L. (2016). One-Shot Video Object Segmentation. *ArXiv:1611.05198 [Cs]*. Retrieved from http://arxiv.org/abs/1611.05198

Caelles, S., Montes, A., Maninis, K.-K., Chen, Y., Van Gool, L., Perazzi, F., & Pont-Tuset, J. (2018). The 2018 DAVIS Challenge on Video Object Segmentation. Retrieved from https://arxiv.org/abs/1803.00557

Camilus, K., & V K, G. (2012). A Review on Graph Based Segmentation. *International Journal of Image, Graphics and Signal Processing*, *4*. https://doi.org/10.5815/ijigsp.2012.05.01

Chen, J., Yang, L., Zhang, Y., Alber, M., & Chen, D. Z. (2016). Combining Fully Convolutional and Recurrent Neural Networks for 3D Biomedical Image Segmentation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 29* (pp. 3036–3044). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/6448-combining-fully-convolutional-and-recurrent-neural-networks-for-3d-biomedical-image-segmentation.pdf

Cheng, J., Liu, S., Tsai, Y.-H., Hung, W.-C., De Mello, S., Gu, J., … Yang, M.-H. (2017). Learning to Segment Instances in Videos with Spatial Propagation Network.

Cheung, S. S., & Kamath, C. (2007). *Robust techniques for background subtraction in urban traffic video*.

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *ArXiv:1511.07289 [Cs]*. Retrieved from http://arxiv.org/abs/1511.07289

Cohen, J., (1988). *Statistical power analysis for the behavioural sciences (2nd Ed.).*

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., … Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. *ArXiv:1604.01685 [Cs]*. Retrieved from http://arxiv.org/abs/1604.01685

Csurka, G., Larlus, D., & Perronnin, F. (2013). What is a good evaluation measure for semantic segmentation? In *Procedings of the British Machine Vision Conference 2013* (pp. 32.1-32.11). Bristol: British Machine Vision Association. https://doi.org/10.5244/C.27.32

Cucchiara, R., Grana, C., Piccardi, M., & Prati, A. (2003). Detecting Moving Objects, Ghosts, and Shadows in Video Streams. *IEEE Trans. Pattern Anal. Mach. Intell.*, *25*(10), 1337–1342. https://doi.org/10.1109/TPAMI.2003.1233909

Cutler, R., & Davis, L. S. (1998). View-based detection and analysis of periodic motion. In *ICPR*. https://doi.org/10.1109/ICPR.1998.711189

Endres, I., & Hoiem, D. (2010). Category Independent Object Proposals. In *ECCV*. https://doi.org/10.1007/978-3-642-15555-0_42

Erdem, Ç. E., Ernst, F., Redert, A., & Hendriks, E. (2005). Temporal stabilization of Video Object Segmentation for 3D-TV applications. *Signal Processing: Image Communication*, *20*(2), 151–167. https://doi.org/10.1016/j.image.2004.10.005

Ess, A., Mueller, T., Grabner, H., & Gool, L. V. (2009). Segmentation-Based Urban Traffic Scene Understanding. In *BMVC*. https://doi.org/10.5244/C.23.84

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). *The PASCAL Visual Object Classes (VOC) challenge*.

Fan, Q., Zhong, F., Lischinski, D., Cohen-Or, D., & Chen, B. (2015). JumpCut: Non-successive Mask Transfer and Interpolation for Video Cutout. *ACM Trans. Graph.*, *34*(6), 195:1–195:10. https://doi.org/10.1145/2816795.2818105

Field, A., Mile, J., Field, Z. (2012). Discovering statistics using R (2nd Ed.).

Friedman, N., & Russell, S. (1997). Image Segmentation in Video Sequences: A Probabilistic Approach. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* (pp. 175–181). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2074226.2074247

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A Review on Deep Learning Techniques Applied to

Semantic Segmentation. *ArXiv:1704.06857 [Cs]*. Retrieved from http://arxiv.org/abs/1704.06857

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., & Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, *70*, 41–65. https://doi.org/10.1016/j.asoc.2018.05.018

Giordano, D., Murabito, F., Palazzo, S., & Spampinato, C. (2015). Superpixel-based video object segmentation using perceptual organization and location prior. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4814–4822). Boston, MA, USA: IEEE. https://doi.org/10.1109/CVPR.2015.7299114

Girshick, R. (2015). Fast R-CNN. *ArXiv:1504.08083 [Cs]*. Retrieved from http://arxiv.org/abs/1504.08083

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *ArXiv:1311.2524 [Cs]*. Retrieved from http://arxiv.org/abs/1311.2524

Grundmann, M., Kwatra, V., Han, M., & Essa, I. (2010). Efficient hierarchical graph-based video segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2141–2148). https://doi.org/10.1109/CVPR.2010.5539893

Gu, C. (2009). Recognition using regions. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 1030–1037. https://doi.org/10.1109/CVPRW.2009.5206727

Guo, Y., Liu, Y., Georgiou, T., & S. Lew, M. (2017). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*. https://doi.org/10.1007/s13735-017-0141-z

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *ArXiv:1703.06870 [Cs]*. Retrieved from http://arxiv.org/abs/1703.06870

He, K., & Sun, J. (2014). Convolutional Neural Networks at Constrained Time Cost. *ArXiv:1412.1710 [Cs]*. Retrieved from http://arxiv.org/abs/1412.1710

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *ArXiv:1512.03385 [Cs]*. Retrieved from http://arxiv.org/abs/1512.03385

Heikkilä, J., & Silvén, O. (1999). A Real-Time System for Monitoring of Cyclists and Pedestrians. In *Proceedings of the Second IEEE Workshop on Visual Surveillance* (pp. 74–). Washington, DC, USA: IEEE Computer Society. Retrieved from http://dl.acm.org/citation.cfm?id=832292.836164

Hu, Y.-T., Huang, J.-B., & Schwing, A. G. (2018). MaskRNN: Instance Level Video Object Segmentation. *ArXiv:1803.11187 [Cs]*. Retrieved from http://arxiv.org/abs/1803.11187

Huang, T. S. (1996). Computer Vision: Evolution and Promise, 5.

Jain, S. D., Xiong, B., & Grauman, K. (2017). FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *ArXiv:1701.05384 [Cs]*. Retrieved from http://arxiv.org/abs/1701.05384

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and Understanding Recurrent Networks. *ArXiv:1506.02078 [Cs]*. Retrieved from http://arxiv.org/abs/1506.02078

Khoreva, A., Benenson, R., Ilg, E., Brox, T., & Schiele, B. (2017). Lucid Data Dreaming for Multiple Object Tracking. *ArXiv:1703.09554 [Cs]*. Retrieved from http://arxiv.org/abs/1703.09554

Khoreva, A., Perazzi, F., Benenson, R., Schiele, B., & Sorkine-Hornung, A. (2016). Learning Video Object Segmentation from Static Images. *ArXiv:1612.02646 [Cs]*. Retrieved from http://arxiv.org/abs/1612.02646

Le, T.-N., Nguyen, K.-T., Nguyen-Phan, M.-H., Ton, T.-V., Trinh, X.-S., Dinh, Q.-H., … Tran, M.-T. (n.d.). Instance Re-Identification Flow for Video Object Segmentation, 6.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

LeCun, Yann, Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, Y. J., Kim, J., & Grauman, K. (2011). Key-segments for Video Object Segmentation. In *Proceedings of the 2011 International Conference on Computer Vision* (pp. 1995–2002). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/ICCV.2011.6126471

Li, F., Kim, T., Humayun, A., Tsai, D., & Rehg, J. M. (2013). Video Segmentation by Tracking Many Figure-Ground Segments. In *2013 IEEE International Conference on Computer Vision* (pp. 2192–2199). Sydney, Australia: IEEE. https://doi.org/10.1109/ICCV.2013.273

Li, X., Qi, Y., Wang, Z., Chen, K., Liu, Z., Shi, J., … Loy, C. C. (2017). Video Object Segmentation with Re-identification. *ArXiv:1708.00197 [Cs]*. Retrieved from http://arxiv.org/abs/1708.00197

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). Feature Pyramid Networks for Object Detection. *ArXiv:1612.03144 [Cs]*. Retrieved from http://arxiv.org/abs/1612.03144

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., … Dollár, P. (2014). Microsoft COCO: Common Objects in Context. *ArXiv:1405.0312 [Cs]*. Retrieved from http://arxiv.org/abs/1405.0312

Long, J., Shelhamer, E., & Darrell, T. (2014). Fully Convolutional Networks for Semantic Segmentation. *ArXiv:1411.4038 [Cs]*. Retrieved from http://arxiv.org/abs/1411.4038

Maas, A. L. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models.

Maninis, K.-K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., & Van Gool, L. (2017). Video Object Segmentation Without Temporal Information. *ArXiv:1709.06031 [Cs]*. Retrieved from http://arxiv.org/abs/1709.06031

McFarlane, N. J. B., & Schofield, C. P. (1995). Segmentation and tracking of piglets in images. *Machine Vision and Applications*, *8*(3), 187–193. https://doi.org/10.1007/BF01215814

Newswanger, A. (2017). One-Shot Video Object Segmentation with Iterative Online Fine-Tuning.

Ochs, P., Malik, J., & Brox, T. (2014). Segmentation of Moving Objects by Long Term Video Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, *36*(6), 1187–1200. https://doi.org/10.1109/TPAMI.2013.242

Panboonyuen, T., Jitkajornwanich, K., Lawawirojwong, S., Srestasathiern, P., & Vateekul, P. (2017). Road Segmentation of Remotely-Sensed Images Using Deep Convolutional Neural Networks with Landscape Metrics and Conditional Random Fields. *Remote Sensing*, *9*, 680. https://doi.org/10.3390/rs9070680

Panboonyuen, T., Vateekul, P., Jitkajornwanich, K., & Lawawirojwong, S. (2018). An Enhanced Deep Convolutional Encoder-Decoder Network for Road Segmentation on Aerial Imagery (pp. 191–201). https://doi.org/10.1007/978-3-319-60663-7_18

Pedamonti, D. (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task. *ArXiv:1804.02763 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1804.02763

Peng, B., Zhang, L., & Zhang, D. (2013). A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, *46*(3), 1020–1038. https://doi.org/10.1016/j.patcog.2012.09.015

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., & Sorkine-Hornung, A. (2016). A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation (pp. 724–732). Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Retrieved from http://openaccess.thecvf.com/content_cvpr_2016/html/Perazzi_A_Benchmark_Dataset_CVPR_2016_paper.html

Piccardi, M. (2004). Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat.*

No.04CH37583) (Vol. 4, pp. 3099–3104 vol.4). https://doi.org/10.1109/ICSMC.2004.1400815

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., & Van Gool, L. (2017). The 2017 DAVIS Challenge on Video Object Segmentation. *ArXiv:1704.00675 [Cs]*. Retrieved from http://arxiv.org/abs/1704.00675

Price, B. L., Morse, B. S., & Cohen, S. (2009). LIVEcut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *2009 IEEE 12th International Conference on Computer Vision* (pp. 779–786). https://doi.org/10.1109/ICCV.2009.5459293

Remagnino, P., Baumberg, A., Grove, T., Hogg, D., Tan, T., Worrall, A., & Baker, K. (1997). *An Integrated Traffic and Pedestrian Model-Based Vision System*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *ArXiv:1506.01497 [Cs]*. Retrieved from http://arxiv.org/abs/1506.01497

Ren, X., & Malik, J. (2007). Tracking as Repeated Figure/Ground Segmentation. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–8). Minneapolis, MN, USA: IEEE. https://doi.org/10.1109/CVPR.2007.383177

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Retrieved from https://arxiv.org/abs/1505.04597v1

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2014). ImageNet Large Scale Visual Recognition Challenge. *ArXiv:1409.0575 [Cs]*. Retrieved from http://arxiv.org/abs/1409.0575

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Shaban, A., Firl, A., Humayun, A., Yuan, J., Wang, X., Lei, P., … Boots, B. (2017). Multiple-Instance Video Segmentation with Sequence-Specific Object Proposals.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv:1409.1556 [Cs]*. Retrieved from http://arxiv.org/abs/1409.1556

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Highway Networks. *ArXiv:1505.00387 [Cs]*. Retrieved from http://arxiv.org/abs/1505.00387

Stauffer, C., & Grimson, W. E. L. (2000). Learning Patterns of Activity Using Real-Time Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, *22*(8), 747–757. https://doi.org/10.1109/34.868677

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2014). Going Deeper with Convolutions. *ArXiv:1409.4842 [Cs]*. Retrieved from http://arxiv.org/abs/1409.4842

Ter-Sarkisov, A., Ross, R., Kelleher, J., Earley, B., & Keane, M. (2018). Beef Cattle Instance Segmentation Using Fully Convolutional Neural Network. *ArXiv:1807.01972 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1807.01972

Toyama, K., Krumm, J., & al, et. (1999). *Wallflower: Principles and Practice of Background Maintenance*.

Tron, R., & Vidal, R. (2007). A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–8). Minneapolis, MN, USA: IEEE. https://doi.org/10.1109/CVPR.2007.382974

Tsai, D., Flagg, M., Nakazawa, A., & Rehg, J. M. (2012). Motion Coherent Tracking Using Multi-label MRF Optimization. *International Journal of Computer Vision*, *100*(2), 190–202. https://doi.org/10.1007/s11263-011-0512-5

Uijlings, J. R. R., Sande, K. E. A. van de, Gevers, T., & Smeulders, A. W. M. (2013). Selective Search for Object Recognition. *International Journal of Computer Vision*, *104*. Retrieved from https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013

Unger, M., Werlberger, M., Pock, T., & Bischof, H. (2012). Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1878–1885). https://doi.org/10.1109/CVPR.2012.6247887

Valipour, S., Siam, M., Jagersand, M., & Ray, N. (2016). Recurrent Fully Convolutional Networks for Video Segmentation. *ArXiv:1606.00487 [Cs]*. Retrieved from http://arxiv.org/abs/1606.00487

Voigtlaender, P., & Leibe, B. (2017). Online Adaptation of Convolutional Neural Networks for Video Object Segmentation. *ArXiv:1706.09364 [Cs]*. Retrieved from http://arxiv.org/abs/1706.09364

Wang, X. (2015). *Graph based approaches for image segmentation and object tracking* (phdthesis). Ecole Centrale de Lyon. Retrieved from https://tel.archives-ouvertes.fr/tel-01303748/document

Wickham, H. (2016). *ggplot2: elegant graphics for data analysis* (Second edition). Cham: Springer.

Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. (1997). Pfinder: Real-Time Tracking of the Human Body. *IEEE Trans. Pattern Anal. Mach. Intell.*, *19*, 780–785. https://doi.org/10.1109/34.598236

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2016). Aggregated Residual Transformations for Deep Neural Networks. *ArXiv:1611.05431 [Cs]*. Retrieved from http://arxiv.org/abs/1611.05431

Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical Evaluation of Rectified Activations in Convolutional Network. *ArXiv:1505.00853 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1505.00853

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative Study of CNN and RNN for Natural Language Processing. *ArXiv:1702.01923 [Cs]*. Retrieved from http://arxiv.org/abs/1702.01923

Yoon, Y., Jeon, H., Yoo, D., Lee, J., & Kweon, I. S. (2015). Learning a Deep Convolutional Network for Light-Field Image Super-Resolution. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)* (pp. 57–65). https://doi.org/10.1109/ICCVW.2015.17

Yuen, J., Russell, B., Liu, C., & Torralba, A. (2009). LabelMe video: Building a video database with human annotations. In *2009 IEEE 12th International Conference on Computer Vision* (pp. 1451–1458). https://doi.org/10.1109/ICCV.2009.5459289

Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Networks. *ArXiv:1311.2901 [Cs]*. Retrieved from http://arxiv.org/abs/1311.2901

Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011). Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. In *Proceedings of the 2011 International Conference on Computer Vision* (pp. 2018–2025). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/ICCV.2011.6126474

Zhang, R., & Ding, J. (2012). Object Tracking and Detecting Based on Adaptive Background Subtraction. *Procedia Engineering*, *29*, 1351–1355. https://doi.org/10.1016/j.proeng.2012.01.139

Zhou, Q., & Aggarwal, J. K. (2001). Tracking and Classifying Moving Objects from Video.