VECTOR INSTITUTE

Computer Vision: Applications in Manufacturing, Surgery, Traffic, Satellites, and Unlabelled Data Recognition

Vector Institute

Industry Collaborative Project

Technical Report

Prepared by

Vector Project Team (Project Management and AI Engineering Teams), Academic Advisors, Researchers and Industry Sponsors

Participating Industry Sponsors (in alphabetical order)

EY, Intact, Linamar
PwC, RBC, Scotiabank,
Thales, Thomson Reuters

Contributions and Acknowledgements

In alphabetical order by last name

Academic Advisors and Researcher Team Leads

Yalda Mohsanzadeh[4], Frank Rudzicz[3,5]
Leonid Sigal[1], Raghav Goyal[1],
John Jewell[4], Shuja Khalid[3],
Matthew Kowal[2]

Sponsor Participants

Elham Ahmadi (RBC), Andrew Alberts-Scherer (Intact),
Andriy Levitskyy (Scotiabank), Jinbiao Ning (Thales),
Tristan Trim (Linamar), Saeed Pouryazdian (EY),
Sim Sachar, (PwC), Yilei Wu (Scotiabank),
An Zhou (Scotiabank)

Vector Project Team

Sedef Akinli Kocak, Xin Li, Rex Ma,
Shayaan Mehdi, Diana Moyano,
Mazen Al Rifai, Gerald Shen

Other Contributions and Acknowledgements

Ehsan Amjadian (RBC), Gabriel Chan (Hôpital Maisonneuve Rosemont),
Leigh Coop (Linamar), Tarek Elguebaly (EY), Veronica Marin (Thales),
Miti Modi (Intact), Kuldeep Panjwani, Richard Pito (Thomson Reuters), Mark Ryan (Intact),
Stephan Thomas (EY), Lillian Zhen (PwC), Vincent Zha

[1]University of British Columbia, [2]Ryerson University, [3]University of Toronto,
[4]Western University, [5]Vector Institute

EXECUTIVE SUMMARY

The adoption and application of computer vision (CV) is growing in many fields as new deep learning-based architectures deliver greater and more efficient performance on complex image-related tasks. New CV theories and approaches are showing the potential to drive transformative innovation in many industries, including health care, banking, security, transportation, retail, and agriculture. From fully autonomous vehicles to automated clinical diagnosis and surgical support systems, CV systems have a major role in technologies that are ushering in an exciting future.

To accelerate these advances, the Vector Institute and its Industry Sponsor companies collaborated in the Computer Vision Project. Hosted by Vector's Industry Innovation Team, the Computer Vision Project brought 15 Vector researchers together with 14 technical professionals from multiple sponsor companies to explore the application of new CV approaches to tasks inspired by their real-world industry needs. In addition to facilitating innovation, the Project was also designed to foster greater collaboration between academic researchers and industry practitioners, increasing the general proficiency of participants at building end-to-end CV pipelines, and growing ecosystem capacity to drive further advances and application of large-scale CV models.

Project participants, divided into three working groups, designed and performed experiments using three CV approaches: **anomaly and semantic segmentation, two-stream neural networks, and transfer learning.** These approaches were applied in the following five use cases:

1. **Anomaly detection in manufacturing (Section 2):** Participants explored the use of autoencoders trained on the MVTec Anomaly Detection dataset to optimize anomaly detection on the manufacturing line. The group trained four types of autoencoders – a regular autoencoder, a variational autoencoder, a constrained, and a context autoencoder – on images of non-defective manufacturing parts. The autoencoders were then tested on image data of real-world parts, assessing each image at the pixel level for anomalies. Autoencoder-based approaches to anomaly detection have the potential to enhance and automate quality assurance at scale.
2. **Semantic segmentation in aerial and road obstacle imagery (Section 3):** Participants applied semantic segmentation techniques to two image sources: satellite imagery and dash cam footage. Semantic segmentation techniques involve labelling each pixel in an image with a class and grouping classified pixels to identify objects.

   Satellite imagery: The group applied U-Net, UNet++, Fully Convolutional Networks (FCN) and Deeplabv3 models, trained on the SpaceNet Building Detection V2 dataset, to detect and extract the boundaries of buildings that appear in aerial images of cities. Automating feature extraction from map data — a task that primarily relies on manual techniques today — can dramatically increase task efficiency and support important downstream uses, including humanitarian efforts, disaster response, and various industrial and public-sector remote sensing activities.

   Road obstacle detection: The group applied U-Net and FCN, trained on the Lost and Found dataset, to detect road obstacles that appear in vehicle dashcam footage. Highly performant obstacle detection is crucial for the realization of fully autonomous vehicles, which require exceptionally low object detection failure rates in order to meet safety certifications standards.

3. **Automated traffic incident detection with two-stream neural networks (Section 4):** Participants applied two-stream neural networks to dashcam footage to detect frames containing hazards, localize those hazards, and classify them by hazard type. Two-stream neural network architecture comprises two convolutional neural networks — one dedicated to spatial features and the other dedicated to temporal features — and are a key advancement in video understanding. The team trained the YOWO network architecture using the Detection of Traffic Anomaly (DoTA) dataset, a benchmark dataset for video anomaly detection. Insights from the experiment can inform technology development to help mitigate vehicle collisions and other road accidents arising from risky road behaviour.

4. **Identifying clinically-relevant features of interest in cholecystectomy procedures (gallbladder surgery) (Section 5):** Participants applied semantic and instance segmentation techniques to enable real-time identification of specific anatomical regions (e.g., the common bile duct, hepatic artery, and portal vein) that are 'no-go zones' for surgeons performing laparoscopic cholecystectomy (the surgical removal of the gallbladder). Using training, The team applied U-Net and Detectron2 to images from the CholecSeg8k dataset (with no-go zones extracted and annotated by a clinical partner) to detect, segment, and classify these areas along with other features.

5. **Transfer learning for efficient video classification and detection (Section 6):** Participants studied the efficacy of transfer learning for detecting and classifying actions in videos that contain few or zero annotations. Transfer learning is the application of knowledge acquired in one machine learning task to another related machine learning task. Done successfully, transfer learning can significantly reduce data preparation costs and increase training efficiency. The team tested multi-modal similarity measures (lingual and visual) to transfer concepts. They transferred off-the-shelf, pre-trained action classifiers trained for spatio-temporal action detection on the Kinetics dataset to SlowFast model architecture using the AVA dataset.

Full descriptions of the technical implementations and results of each use case are provided in the report.

The Computer Vision Project resulted in significant knowledge transfer between Vector researchers and industry participants, enhancing the real-world application potential of leading-edge computer vision theory and techniques. The project has also revealed promising avenues for future exploration, including self-supervised learning (for general application), auto-annotation and data augmentation techniques (for application in clinical settings), open-set segmentation (for anomaly and semantic segmentation in image data), and techniques for the prevention of over-fitting (for traffic incident detection). Finally, the insights and results achieved in the Project illustrate the importance of collaboration between industry and academia, and make evident the role these joint efforts play in accelerating the innovative industrial application of new developments in artificial intelligence.

CONTENTS

[1] This work has been presented in the Location Intelligence and Knowledge Extraction 2022 Conference

1 INTRODUCTION

The Computer Vision industry collaboration project launched by the Vector Institute (Vector) aimed to explore new approaches in computer vision (CV) in an applied setting with the supervision and expertise of Vector researchers, bringing together theoretical expertise with the deep domain knowledge of Vector sponsor companies.

This report provides an overview of the work that has been completed in collaboration between Vector and industrial sponsors in the project. The project is intended to help Vector sponsors prepare their workforce for new advancements in vision systems and ensure they have the skills and tools to be more resilient as the world continues to evolve. The aim was to explore new applications of CV in the business setting as new theoretical approaches can enable better results compared to traditional methods.

With the development of automated image analysis technology, has been shown to be a promising solution for various challenges that require specialized and labor intensive image analyses, including medical imaging

Recent approaches in computer vision and deep learning have inspired research and applications in decision making efforts and they also have the potential to revolutionize many areas such as autonomous driving, medical diagnostics, finance,and agriculture, and many more. These approaches have led to the development of novel and robust tools such as for medical imaging (e.g., [1, 2, 3, 4]) and for infectious diseases (e.g.,[5, 6]).

1.1 PROJECT OVERVIEW

This is a joint academic-industrial collaborative project launched in the summer of 2019 to explore opportunities as well as promote recent advances in the Computer Vision domain. The project involved 29 participants: 15 Vector researchers and staff with expertise in machine learning and vision systems along with 14 industry technical professionals from Vector sponsor companies, namely Linamar, RBC, Thales Group, PwC, Scotiabank, Intact, and EY.

The participants established three working groups, each of which developed and performed experiments relevant to existing industry needs.

The primary objectives of the project were:

- foster and widen productive collaboration among academic researchers and industry practitioners on projects in the CV domain,
- help participants in gaining proficiency in building an end-to-end pipeline from data ingestion to large scale training and downstream fine-tuning, and
- build the capacity for further advances and new lines of businesses in large-scale vision models in our ecosystem.

The project was conducted over 10 months. Weekly meetings were held to communicate current

updates and tasks among project members. Common group activities in the weekly meetings included problem solving, decision making, prioritization, and task assignment. Weekly meetings also featured invited guest talks, tutorials on recent advances in CV and ML from academia and industry, and reading group activities of recent related literature.

ARegarding high-performance computing resources, Vector Institute provided high performance computing resource with GPUs for development and deployment of large-scale CV architectures. For the model development and evaluation, the implementation from the Vector Computer Vision tools [2] was used. The code for each use case implementation included the model, the metrics, the datasets, the training and testing sets and was stored in a separate repository [3] Further details can be found in the corresponding sections in GitHub.

Three broader topic related approaches were explored: anomaly and semantic segmentation, two-stream neural networks and transfer learning, which were applied to five specific image and video recognition use cases. These use-cases reflected current industry needs, participants' interests and expertise, and opportunities to translate academic advances into real-world applications:

1. Anomaly detection in manufacturing
2. Semantic and instance segmentation in aerial and road obstacle imagery
3. Semantic segmentation in identifying clinically relevant features of interest in cholecystectomy procedures (gallbladder surgery)
4. Automated traffic incident detection With two-stream neural networks
5. Transfer learning for efficient video classification and detection

The remainder of this report provides an overview of each use-case and includes background and objective, dataset and modelling approach, results and discussion, and limitations and best practices.

2 ANOMALY SEGMENTATION IN MANUFACTURING

Contributors: Elham Ahmadi, John Jewell, Jinbiao Ning, Sim Sachar, Tristan Trim

2.1 BACKGROUND/OBJECTIVE

Anomaly detection is an important task in computer vision that is concerned with identifying anomalous images given a training set of only normal images. In anomaly segmentation, the concept of anomaly detection is extended to the pixel level in order to identify anomalous regions of images. There are many applications to anomaly detection including biomedical image segmentation, video surveillance and defect detection. In particular, defect detection involves detecting abnormalities in manufacturing components and so is widely used in the industry to enhance quality

[2] https://github.com/VectorInstitute/vector_cv_tools
[3] https://github.com/VectorInstitute/Computer_Vision_Project

assurance and efficiency in the production process [7]. However, having a person manually inspect each component is not feasible in most cases. To address this, systems have been proposed to automate the detection of defective components. These approaches generally take as input an image of a component and output a label or pixel-level mask that predicts whether the image or pixel is anomalous. Although initial approaches were generally ineffective, newer, deep learning based approaches have shown very strong performance in anomaly detection and segmentation [8]. Thus, these new methods have the potential to dramatically increase quality assurance and efficiency. In order to compare anomaly detection methods, several datasets have been proposed as benchmarks such as MNIST [9], CIFAR [10], and UCSD [11], whereas there are much fewer benchmark datasets for the anomaly segmentation task. To address this, the MVTec Anomaly Detection Dataset [7] was recently introduced as a benchmark for anomaly segmentation.

The goal of this focus area was to apply state-of-the-art methods to accurately segment anomalies in the MVTec dataset. In doing so, we compared the performance of different anomaly segmentation methods in the industrial inspection setting. Additionally, we sought to optimize the performance of the methods by altering the hyperparameters and architectures of the approaches. The approaches and corresponding results will be discussed at length in the following section.

2.2 METHOD

Anomaly segmentation is a challenging task because there are no examples of anomalies available to train a segmentation network to discern between normal and anomalous pixels. As such, methods focus on modelling the distribution of inlier data and checking if new samples conform to the distribution of the inlier data at the pixel level. Conventional methods employ PCA, one class SVM and their variations, to learn a subspace that represents inliers samples well [8]. Unsupervised techniques such as Gaussian mixture models and k-means have also been employed to estimate the distribution of inliers and outliers. Unfortunately, these methods perform poorly when applied to high dimensional data [5].

Recent approaches to anomaly detection employ deep learning to model the distribution of high dimensional data. To this end, deep autoencoders have been a popular choice of architecture which is a neural network that learns to generate low dimensional encodings from which the original input sample can be reconstructed [5]. As an effective approach to generate compressed representations of data without labels, autoencoders are used widely across different modalities of data including images, videos, text and speech. Autoencoders consist of two key components, an encoder and a decoder. The encoder learns a mapping from an image to a lower-dimensional latent space, and the decoder learns a mapping from the latent space back to the original image. In this way, autoencoders are trained in an unsupervised manner by minimizing the error between the original image and the reconstruction.

Several variations of the autoencoder have been proposed, some of which have been shown to perform well in the anomaly detection task. Specifically, a variational autoencoder enforces that the latent space follows a specified distribution. By ensuring the latent space conforms to a specific distribution, variational autoencoders generate more robust representations for downstream tasks than regular autoencoders. Alternatively, context autoencoders learn to reconstruct samples that have had portions of the input sample masked randomly [12].

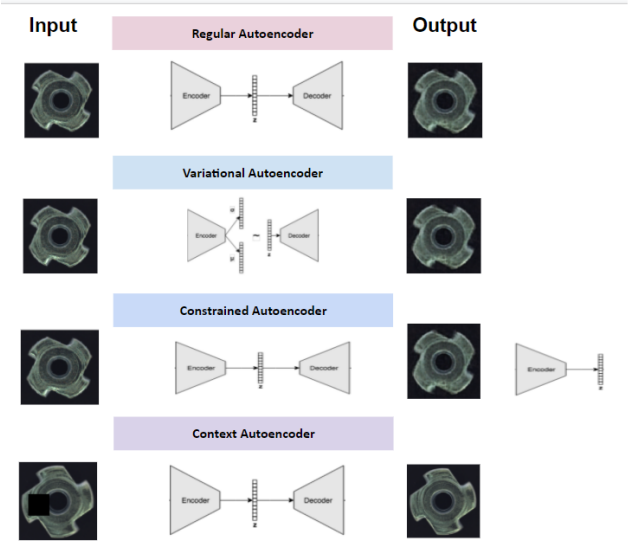


Figure 1: The four autoencoder architectures explored, with the first three having a normal input and the last having a masked input: Regular Autoencoder, Variational Autoencoder, Constrained Autoencoder and Context Autoencoder.

Model: The four autoencoder architectures explored are regular autoencoder, variational autoencoder, constrained autoencoder and context autoencoder with the first three having a normal input and the last having a masked input. An overview of the aforementioned architectures is available in Figure 1.

Dataset: In order to explore the application of autoencoders to the task of anomaly segmentation in manufacturing, the MVTec anomaly detection dataset was used [7]. It contains 5354 high-resolution images from 15 different object categories and includes 70 different types of defects across the anomalous images that are typical in the manufacturing process. For each object category, a training set of normal images of objects and textures as well as a test set with both normal and anomalous samples along with their corresponding labels. An example of the dataset is available in Figure 2.

2.3 RESULTS AND DISCUSSION

Implementation Details: For each autoencoder, the same base architecture was used with a symmetric encoder and decoder. The encoder consists of six convolutional layers with 128, 256, 512, 1024, 2048 and 2048 channels, respectively. A stride size of two and kernel size of four is used to downsample the image to a feature map that is flattened and passed to the decoder. The decoder consists of six transposed convolutional layers with 2048, 2048, 1024, 512, 256 and 128 channels, respectively. A stride size of two and a kernel size of four is used to upsample the input vector to the dimensions of the original image. Each convolutional layer has a kernel size of four and is followed by batch normalization layer and a ReLU non linearity. The models were trained for fifty EPOCHS using the ADAM optimizer with a constant learning rate of between $.001$ and $.0001$. Experiment code was implemented in PyTorch and experiments were conducted on the Vector Institute compute cluster. Each model was trained with a single NVIDIA T4 GPU and took roughly four hours to complete.

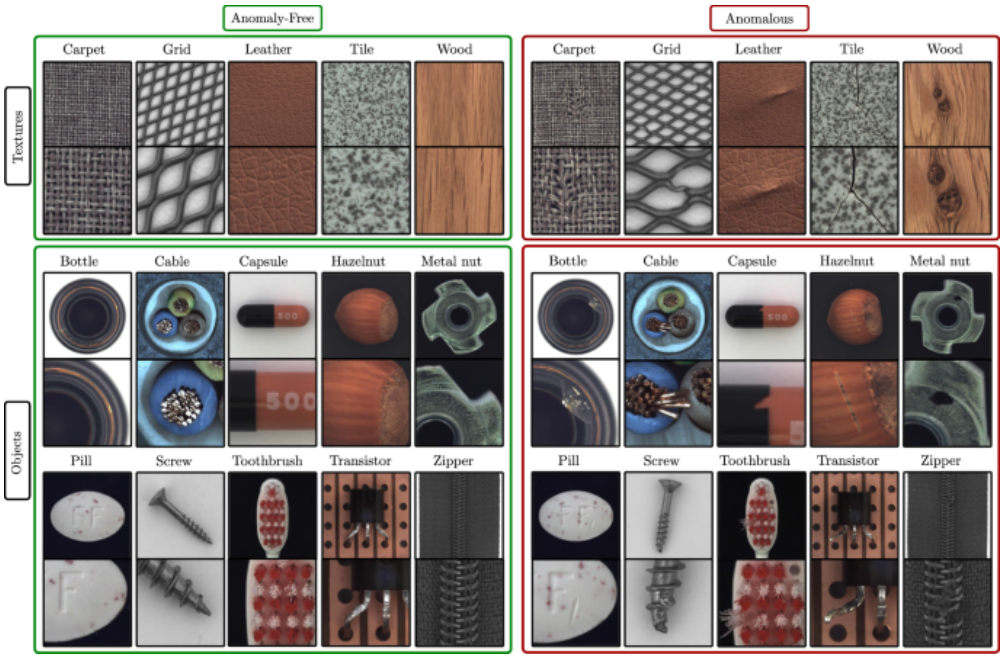


Figure 2: An overview of the MVTec dataset including examples of both normal images and inliers from various object and texture classes.

Experimental Setup: The MVTEC dataset object categories each include a train set of normal samples and a test set of both normal and anomalous samples. Models were optimized to be able to reconstruct samples from the inlier distribution during the training phase. Subsequently, at test time, both normal and anomalous images are input to the model and the pixelwise reconstruction error of samples is used to identify anomalous regions. Specifically, the models were evaluated on the testing data for each of the object categories and the average area under the ROC curve (AUC) is reported. A small validation set of normal images is used to determine which model step yields the most optimal set of parameters. Specifically, 10% of images were randomly removed from the train set and used as the validation set. For testing, the entire test set was used and the average AUC across object categories is reported for each method.

Results: Results from the aforementioned experiments are available in Table 1. Each model achieves a moderate AUC score on the anomaly segmentation task despite being trained with no labels. Although the models perform similarly, it is worth noting that each of the autoencoder variations (context, constrained, variational) outperform the regular autoencoder. This is an indication that these methods have architectures and/or objectives that are better suited to the anomaly detection task than regular autoencoders, a pattern that matches the conclusions of other recent works [5].

The context autoencoder achieves the best performance with an AUC of 0.79065. This strong performance can be attributed to the fact that they offer semantically rich representations by learning to inpaint masked regions of images in addition to reconstructing them. Visual results from the context autoencoder further support the strong quantitative results and are available in Figure 3.

Model	AUC
Context Autoencoder	0.79065
Variational Autoencoder	0.785652
Constrained Autoencoder	0.782606
Regular Autoencoder	0.77138

Table 1: AUC Scores of each model on the MVTEC Dataset.

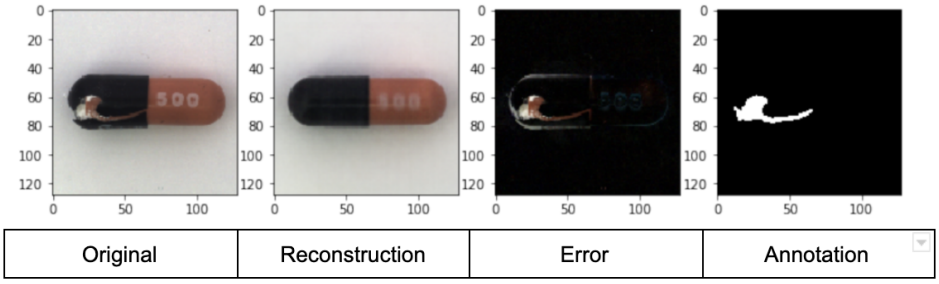


Figure 3: An example of an image, reconstruction, error map and ground truth segmentation mask for an anomalous sample in the MVTEC dataset.

2.4 LIMITATIONS AND BEST PRACTICES

This study offers an applied survey of several variations of autoencoders for the anomaly segmentation task on the MVTec dataset. Although this analysis is helpful for practitioners looking to deploy autoencoder-based anomaly segmentation systems, it does have limitations. Namely, two assumptions were made about the nature of the data. First, it is assumed that the training data consists of only inliers. This may not always be a realistic assumption; especially in the case of large uncurated datasets. In practice, as long as the proportion of anomalies in the training set is low, autoencoder-based approaches to anomaly segmentation will still work well. However, a formal analysis of this is omitted because it is beyond the scope of this project. Second, it is assumed that a small validation set consisting of both inliers and outliers is available to use as a stopping criterion during the training phase. This may not be possible in situations where anomalies are not well defined or extremely hard to sample. In practice, it seems reasonable to amass a small collection of both normal images and anomalous images to use for this purpose but the validity of the assumption varies across use cases. However, in the absence of outliers, other metrics defined over a validation of exclusively normal images can be used as a heuristic for when to stop training.

In light of the aforementioned limitations, we still strongly believe that autoencoder-based approaches offer a robust solution to the anomaly segmentation task, especially when compared to supervised methods that are costly and time consuming to deploy.

3 SEMANTIC SEGMENTATION IN SATELLITE IMAGERY AND ROAD OBSTACLE DETECTION

Contributors: Elham Ahmadi, John Jewell, Jinbiao Ning, Tristan Trim

3.1 SEMANTIC SEGMENTATION OVERVIEW

Semantic segmentation is a subclass of image segmentation where pixels are grouped together based on their class [13]. It plays a critical role in a broad range of applications such as autonomous driving (e.g. self-driving cars or autonomous trains), geospatial analysis (e.g. building footprint extraction) and medical image segmentation (e.g. biomedical marker discovery). The goal of semantic segmentation is to label each pixel of an image with a class, effectively partitioning the pixels in the image into groups based on object type. Due to the high dimensional nature of both the input and the output space, semantic segmentation has traditionally been a very challenging task in computer vision [13]. Fortunately, recent supervised deep learning approaches have achieved robust semantic segmentation performance on a variety of challenging benchmarks [14]. These approaches use large datasets of images with corresponding pixel wise labels to train neural networks by iteratively updating the parameters of the model to minimize a differentiable loss that characterizes the difference between predictions and labels. At inference, new samples are fed to the network and it produces a segmentation map with the same spatial resolution as the input image that encodes the label of each pixel. Figure 4 provides an example of semantic segmentation, where the goal is to predict class labels (i.e. Person, Bicycle, and Background) for each pixel in the image.

The objective of this focus area is to apply cutting edge semantic segmentation methods to a variety of datasets. In particular, we first explore binary semantic segmentation on the Spacenet dataset. The goal is to segment building footprints from the background area in satellite imagery. Next, building on our experience with binary semantic segmentation, we explore the multi-class setting using the Lost and Found dataset. The goal is to segment drivable area, non-drivable area and obstacles in scenes captured from the dashboard camera of a vehicle. A thorough outline of these use cases, along with results from our experiments, will be available in Section 3.2 and 3.3, respectively.

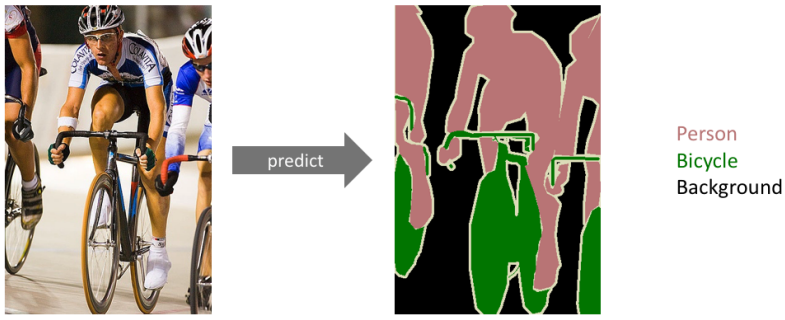


Figure 4: An example of semantic segmentation, where the goal is to predict class labels for each pixel in the image. [15]

3.2 SATELLITE IMAGERY WITH SEMANTIC SEGMENTATION MODELS[4]

3.2.1 BACKGROUND/OBJECTIVE

The objective of this use case is to extract building boundaries from aerial imagery by using advanced semantic segmentation methods. Current approaches to extracting features from maps such as roads, building footprints, and points of interest are primarily based on manual techniques. Advancing automated feature extraction techniques will serve important downstream uses of map data including humanitarian, disaster response and agriculture. Furthermore, solving this challenge is an important stepping stone to unleashing the power of advanced computer vision algorithms on a variety of remote sensing data applications in both the public and private sector.

3.2.2 METHOD

Model: The four approaches to semantic segmentation were explored include: U-Net [14] (Figure 5(a)), U-Net++ [16] (Figure 5 (b)), Fully Convolution Networks (FCN) [17] and Deeplabv3 [18]. For both FCN and Deeplabv3, two variants of the architecture with different backbones (Resnet-50 and Resnet-100) are included. FCN and Deeplabv3 with a Resnet-50 backbone (FCN-50 and DLV-50), a Resnet-101 backbone (FCN-101 and DLV-101) were benchmarked. The backbones were pretrained using the COCO train2017 semantic segmentation dataset [19] and fine-tuned for the building footprint extraction task. In total, six approaches are benchmarked on the task of building footprint extraction in aerial images.

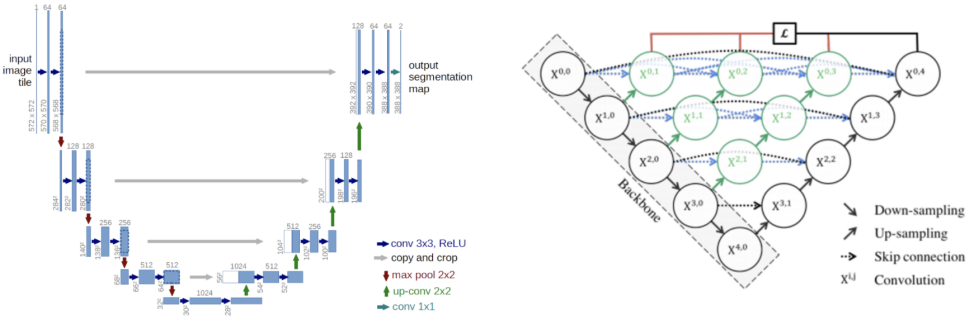
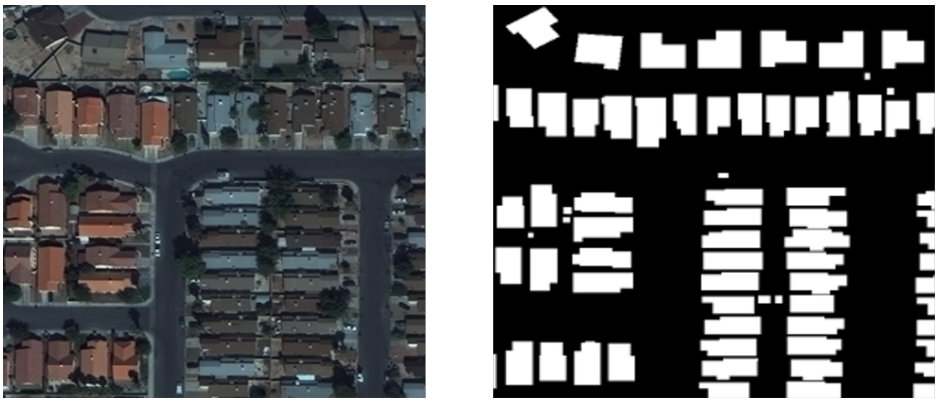


Figure 5: U-Net Architecture [14](a) and UNet++ Architecture [16](b).

Dataset: To explore the application of semantic segmentation to building footprints from satellite imagery the SpaceNet Building Detection V2 dataset [20] was used. This dataset contains 302701 building labels (polygons) from 10,593 multispectral satellite images of Las Vegas, Paris, Shanghai, and Khartoum. The dataset is split into two classes: **building** and **background**, as can be seen in Figure 6.

[4]This work has been presented in the Location Intelligence and Knowledge Extraction 2022 Conference



a) An example of SpaceNet Images

b) The ground truth with two labels: building and background

Figure 6: An example of the images (a) and labels (b) in the Spacenet Building Detection V2. [20]

3.2.3 RESULTS AND DISCUSSION

Implementation Details: The experiments were implemented in Python using the PyTorch Framework and conducted on 4 NVIDIA Telsa P100 GPUs. The architecture for each approach is consistent with that specified in the original papers [14, 16, 17, 18]. Each method is trained for 30 Epochs using the ADAM optimizer with a learning rate of 2e-4. Random seeds are used to strive for consistency in evaluation and reproducibility of the experiments.

Experimental Setup: The dataset is divided into training (80%), validating (10%) and testing (10%) sets. Images are resized from 650x650 to 384x384 using bi-cubic interpolation and normalized using the mean and standard deviation of the Imagenet dataset [21]. The proposed semantic segmentation models are trained on the training set, while the validating set is used to determine a stopping criteria. Lastly, the trained model is evaluated on the testing set. Intersection over Union (IoU) is the metric used to evaluate the model performance and measures the overlap between the labels of the prediction and ground truth. IoU ranges from 0 to 1 where 1 denotes perfect and complete overlap.

Results: The IoU of each method on the test set is reported in Figure 2. DLV3-101 achieves the best performance with an IoU of 0.7734 followed closely DLV3-50, FCN-50 and FCN-101. U-Net and U-Net++ perform comparatively worse with an IoU of 0.5644 and 0.6554, respectively. The performance gap can be attributed to the fact that FCN-50, FCN-101, DLV3-50 and DLV3-100 benefit from pre-training whereas U-Net and U-Net++ do not. This performance gap is also apparent in Figure 8 which shows the train and validation loss of each method across epochs. Methods that leverage pre-training are able to achieve better performance on both the train and validation set from the onset of training. The validation loss begins to plateau after only a few epochs which suggest that training is finished and should be early stopped to prevent overfitting. Alternatively, U-Net and U-Net++ have train and validation losses that consistently decrease over the course of training. This highlights the fact that models that leverage pretraining converge to the optimal set of parameters faster in addition to offering better performance.

Qualitative results are available in Figure 7, which shows an example input image, ground truth label and predicted semantic map for each method. The prediction quality of methods follow

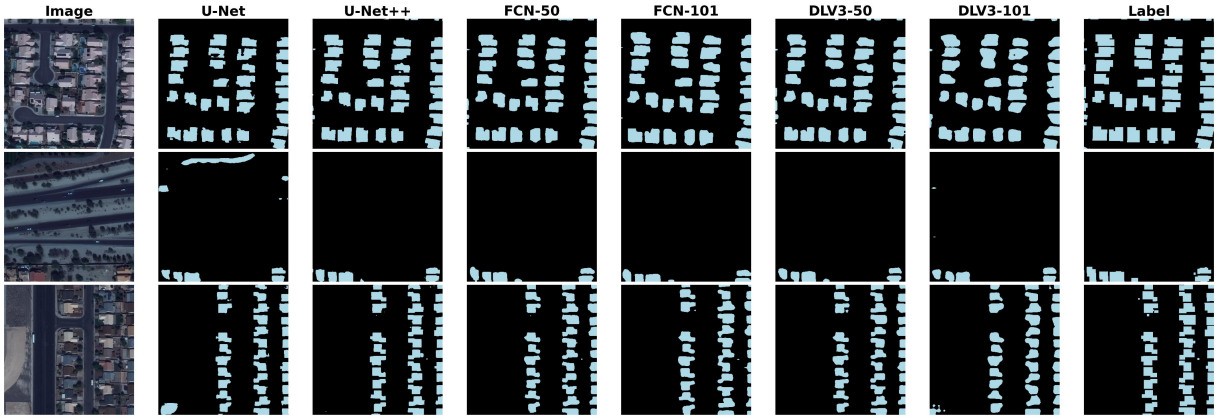


Figure 7: A visualization of the predictions generated by each approach along with the input image (far left) and ground truth label (far right).

the quantitative results but performance is impressive across the board. The methods are able to generate precise semantic maps in scenes densely populated with building footprints. Additionally, predicted semantic maps in scenes that are sparsely populated with building footprints are robust to false positives, even in cases where roadways, parking lots or other structures are present.

A preliminary analysis of the importance of model architecture conditioned on pretraining yields interesting results. The performance among methods that leverage pretraining is similar, even across different architectures and backbones. Conversely, when considering the performance among methods that do not leverage pretraining, U-Net++ vastly outperforms U-Net. Although this warrants further experiments to validate, one hypothesis is that model architecture becomes less relevant as the amount of pretraining increases.

MODEL	U-NET	UNET++	FCN-50	FCN-101	DLV3-50	DLV3-101
UoI	0.5644	0.6554	0.7455	0.7472	0.7612	0.7734

Table 2: IoU score on test set for each approach

3.3 ROAD OBSTACLE DETECTION WITH CNN AND U-NET

3.3.1 BACKGROUND/OBJECTIVE

Detecting obstacles on the road/railway is a critical part of the driving task which has not been mastered by fully autonomous vehicles. Semantic segmentation plays an important role in addressing the challenges of identifying the locations of obstacles. The dataset is divided into training (80%), validating (10%) and testing (10%) sets. Images are resized from 650x650 to 384x384 using bi-cubic interpolation and normalized using the mean and standard deviation of the Imagenet dataset [21]. The proposed semantic segmentation models are trained on the training set, while the validating set is used to determine a stopping criteria. Lastly, the trained model is evaluated on the testing set. Intersection over Union (IoU) is the metric used to evaluate the model performance and measures the overlap between the labels of the prediction and ground truth. IoU ranges from 0 to 1 where 1 denotes perfect and complete overlap with obstacles on the road and railway. There-

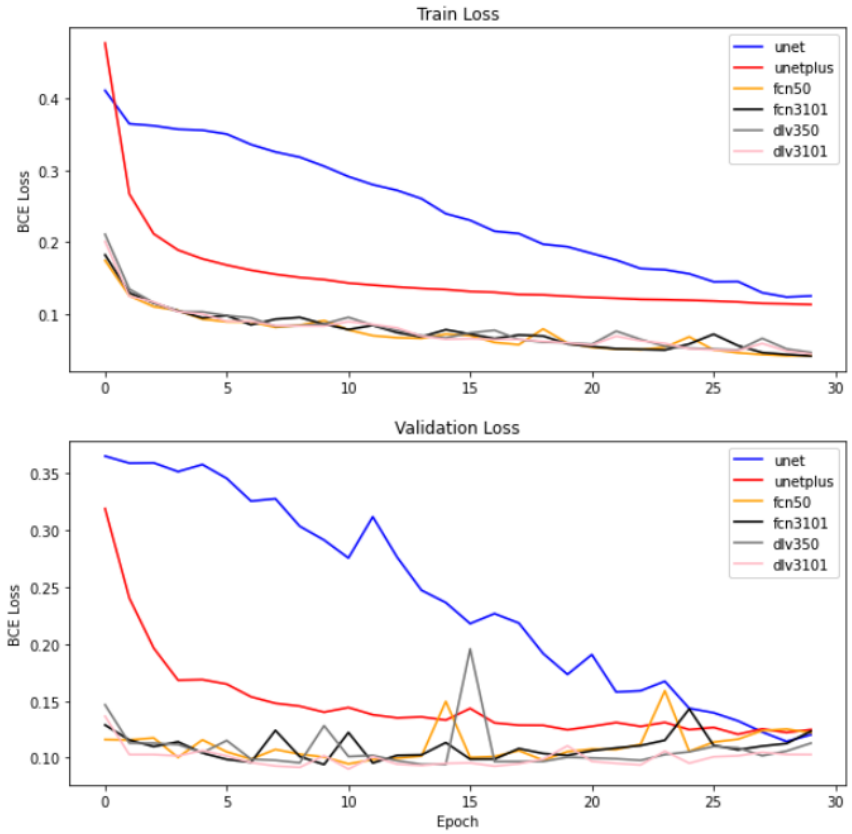


Figure 8: Binary cross entropy loss for training set (top) and validation set (bottom) across epochs.

fore, the second use case we chose to explore was semantic segmentation methods for obstacle detection on the road and railway.

3.3.2 METHOD

Model: The two models we have chosen to explore are popular semantic segmentation models: U-Net [14] and fully convolutional networks (FCN) [17]. Since U-Net was introduced in section 3.2.2, the following section offers a brief description of the FCN architecture only.

FCN efficiently learn to make dense predictions for per-pixel tasks [17]. They are trained end-to-end to perform semantic segmentation by mapping arbitrary-sized input images to predicted semantic maps using convolutional layers. In-network upsampling layers are leveraged to make pixel-wise predictions by increasing the spatial resolution of the features generated by the backbone of the network to the desired height and width of the output semantic map. To this end, contemporary classification networks (such as AlexNet [22], VGG net [23], and ResNet [24]) are often used as backbones and transfer their learned representations by fine-tuning [17] to the segmentation task. The FCN used in our experiments has an architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentation by using skip connections. An overview of the FCN architecture is available in Figure 9.

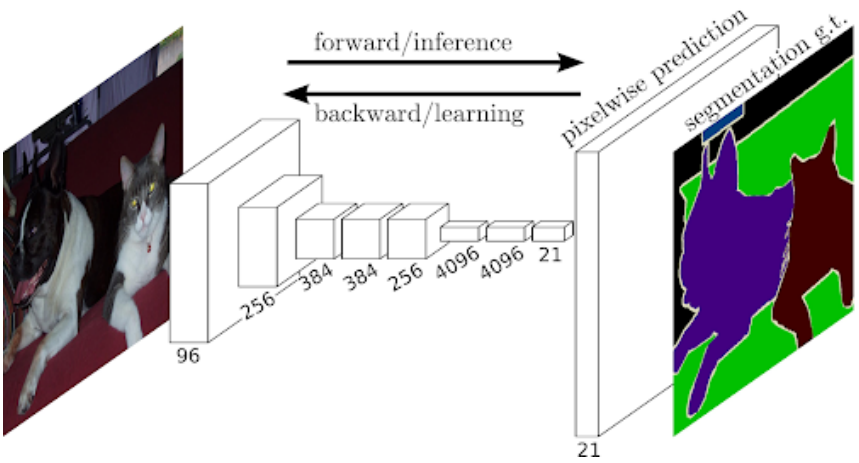
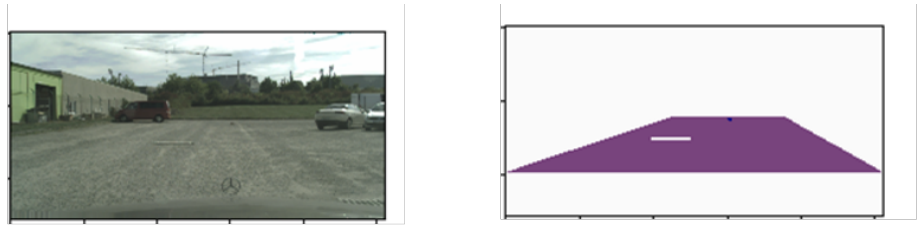


Figure 9: The architecture of FCN [17].

Dataset: In order to explore the application of semantic segmentation to detect road obstacles for autonomous vehicles, the Lost and Found dataset [25] was used. It was introduced to evaluate the performance of small road obstacle detection approaches [25]. The Lost and Found Dataset includes 2k images recording from 13 different challenging street scenarios, featuring 37 different obstacles types. Each object is labeled with a unique ID, allowing for a later refinement into sub-categories. An overview of the Lost and Found dataset is available in Figure 10, which is refined into three classes: driveable area, non drivable area and obstacles.



a) An example of Lost and Found dataset

b) The ground truth with three labels: driveable area, non drivable area and obstacle

Figure 10: An example image from the Lost and Found Dataset (a) and the ground truth with three labels (b).

3.3.3 RESULTS AND DISCUSSION

Implementation Details: U-Net was implemented as a baseline and compared with FCN. The cross entropy coefficient was monitored and an early-stopping mechanism was applied to the validation set. The network was optimized using the Adam optimizer with a learning rate of 1e-5.

Experimental Setup: The labels in the Lost and Found dataset were refined into 3 Classes: driveable area, non drivable area and obstacles. The Lost and Found dataset was divided into training (80%), validating (10%) and testing (10%) set. The proposed semantic segmentation models were trained on the training set, while the validating set was used to determine the criteria and the

trained model was tested by the testing set.

Results: Although both approaches perform similarly, the pre-trained FCN achieves the best performance on the validation set as evidenced by the quantitative results in Figure 11 and the qualitative comparison in Figure 12. In addition, FCN is also more stable to train. This most likely stems from the fact that FCN uses a Resnet 50 pretrained backbone while U-Net is trained entirely from scratch.

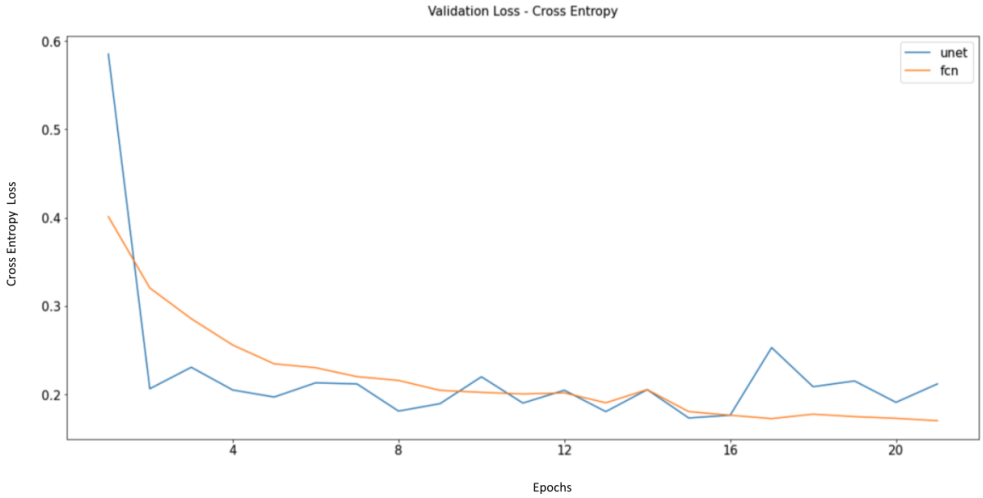


Figure 11: Cross Entropy Loss on Validation Set for U-Net and FCN across epochs.

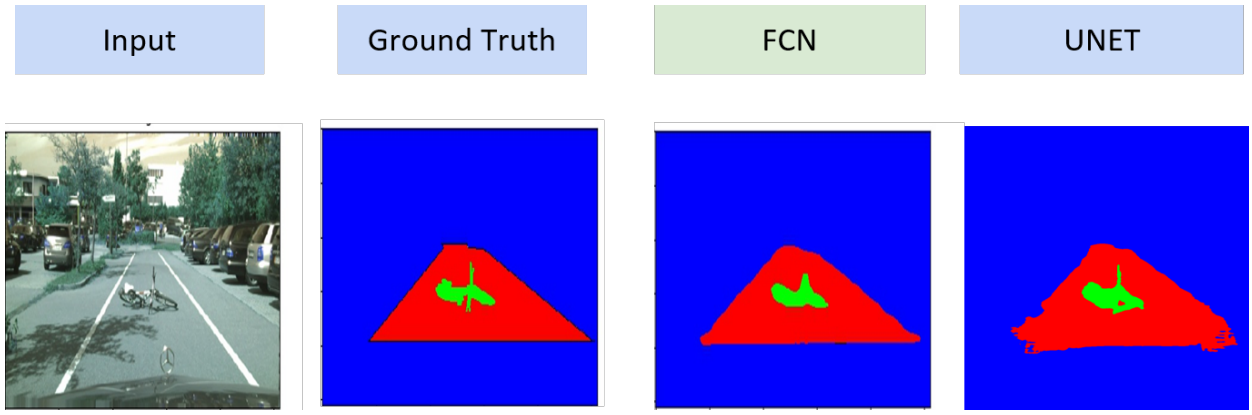


Figure 12: Qualitative comparison between U-Net and FCN.

3.4 LIMITATIONS AND BEST PRACTICES

Real-time Consideration: The inference time is critical in obstacle detection for autonomous vehicles. However, U-Net, UNet++ and FCN are not a real-time detectors. Further exploration on real-time model architectures is important (e.g. ICNET [26]) to consider when deploying production systems with low-latency requirements.

Supervision and Labelling: Semantic segmentation is a supervised learning method. This im-

plies that it requires a dataset of images along with corresponding pixel-wise labels. In the case of anomalous objects, it is difficult to obtain labelled examples by virtue of the object being out-of-distribution. Accordingly, it may not always be realistic to obtain labelled examples of object classes that are out-of distribution. Thus, in the absence of labelled examples for some classes, recent approaches have resorted to incorporating anomaly segmentation techniques into supervised semantic segmentation methods [27, 28]. In cases where labels are not available for a certain subset of classes, these approaches are more suitable.

4 AUTOMATED TRAFFIC INCIDENT DETECTION WITH TWO-STREAM NEURAL NETWORKS[1]

Contributors: Andrew Alberts-Scherer, Matthew Kowal

4.1 BACKGROUND/OBJECTIVE

Being able to detect risky road behaviour is an important step in the mitigation of vehicle collisions and other road accidents. For example, drivers demonstrating reckless driving or near-misses may be helped by additional road training to correct behaviours and prevent future collisions. Knowledge of particular risky driving maneuvers, such as swerving and tailgating can also be of use for insurance companies in better underwriting and pricing auto insurance policies, thus allowing them to be more selective and more fair in their determination of premiums.

One tool to achieve this is the dashcam; a continuously recording front-facing camera aimed out the windshield of a car. Footage from these cameras have been used to determine the at-fault party in accidents, and as of more recently are being used as part of risk-mitigation programs for commercial fleets. Fleet managers use these tools to ensure compliance of road safety and to discover drivers in need of additional training [29].

Making use of large quantities of dashcam footage is however difficult in a manual fashion. For example, a salaried professional would need to watch through hundreds of thousands of hours of video to pinpoint where risks happen and what those risks are. A more tractable approach is to harness techniques from computer vision to produce an artificially intelligent system which looks at all the videos and automatically detects the frame or set of frames involving a hazard and classifies the type of risk and localizes the subjects of the risk.

4.2 METHOD

4.2.1 DATASET

The Detection of Traffic Anomaly (DoTA) dataset [30] is a benchmark dataset used for video anomaly detection. It uses a where-what-when labelling scheme, where the objective is to detect, localize, and recognize traffic incidents or anomalies. The dataset contains 4,677 videos (731,932 frames) at 10 frames per second (fps) with 1280x720 resolution. The dataset is collected from YouTube and has nine anomaly classes (see Table 1). Every frame is either labelled as an anomaly or a non-anomalous frame. The anomalous frames have two types of labels (i) bounding boxes of each object in the anomaly with corresponding object category labels and (ii) the anomaly class. There-

Anomaly ID	Description of Anomaly	# of Videos in Original / Ours
1	Collision with another vehicle which starts, stops, or is stationary	95 / 89
2	Collision with another vehicle moving ahead or waiting	663 / 601
3	Collision with another vehicle moving laterally in the same direction	726 / 618
4	Collision with another oncoming vehicle	477 / 427
5	Collision with another vehicle which turns into or crosses a road	1696 / 1506
6	Collision between vehicle and pedestrian	100 / 92
7	Collision with an obstacle in the roadway	95 / 53
8	Out-of-control and leaving the roadway to the left or right	732 / 459
9	Unknown	92 / 55

Table 3: The nine types of anomalies in the DoTA dataset. The dataset is downloaded from YouTube and therefore some of the videos are unavailable. We show the number of videos in the original dataset and our version in the final column.

Object ID	Description of Object	# of Occurrences in Train + Val set
1	Person	61+31
2	Rider	284+111
3	Car	2978+1264
4	Bus	55+41
5	Truck	375+160
6	Bike	29+13
7	Motorcycle	243+88

Table 4: The object distribution in the DoTA dataset.

fore, the task of the network is to predict whether or not the frame is anomalous, and if it is, detect the objects involved in the incident and also classify the anomaly. The dataset has a diverse range of scenes and the number of examples per class vary significantly, which makes it a challenging task. For example, the smallest class (9: Unknown) has 92 videos while the largest class (5: Collision with another vehicle which turns into or crosses a road) has 1696.

4.2.2 MODEL

The network architecture we chose for this problem is ”you Only Watch Once” (YOWO) [31]. While there are many video-detection architectures we could have selected, this architecture was chosen because (i) it is efficient and easy to implement and (ii) it is modular.

The default YOWO architecture (see Fig. 1) is a modular architecture designed for action detection in video. YOWO has three main components: (i) the 3D CNN backbone, the (ii) 2D CNN backbone, and (iii) the Channel Fusion and Attention Module (CFAM). The 3D CNN backbone takes as input 16 frames from a video and the 2D CNN backbone takes in the current frame as input. Both networks output a feature representation with the same spatial size, with the 3D CNN output having the temporal dimension average pooled. The CFAM module takes both representations as input, and performs a channel-wise attention between them. The model uses both a 3D and 2D input to allow for the network to learn to use both motion and appearance visual information in each stream. The CFAM module then allows for a global comparison of these features along the channel dimension to determine which features are required for the final bounding box predictions.

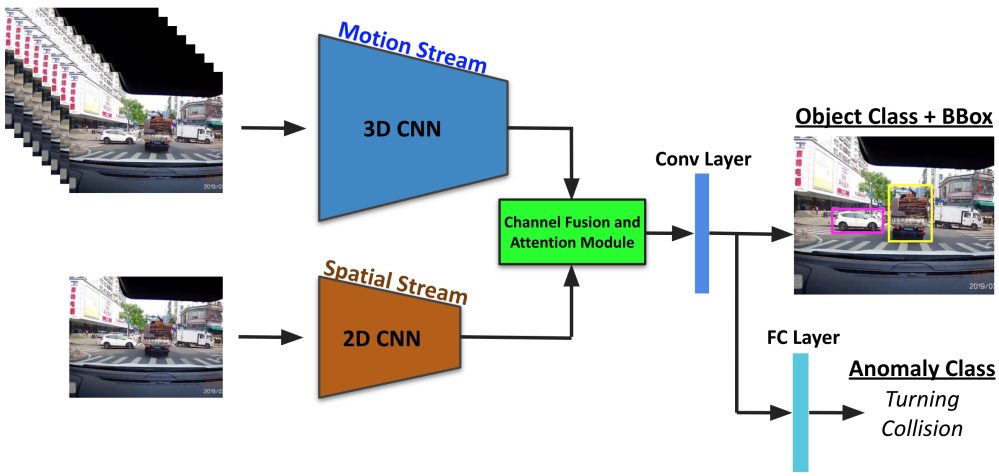


Figure 13: Our modified version of the YOWO architecture. Our modification is the addition of the fully connected (FC) layer which predicts an anomaly classification along with the bounding box predictions.

The output of the network uses anchor boxes to determine the final bounding box predictions. Anchor boxes are precomputed bounding boxes (i.e., a height and a width), which are copied many times over the entire image. Bounding box proposals are obtained from the final feature representation of the YOWO model. Then non-maximum suppression is applied to remove noisy or duplicate bounding boxes from the proposals. The remaining boxes are then compared (i.e., overlap) with all of the anchor boxes and only the ones with the most overlap are kept as the final prediction.

Architectural changes. The first architectural change we made was to replace the backbone networks with ones pre-trained on more similar tasks as ours (traffic anomaly detection). The default architecture uses a DarkNet [32] trained on PASCAL-VOC [33] for the 2D CNN and a ResNeXt-101 [34] trained on Kinetics400 [35] for the 3D CNN. PASCAL-VOC has some classes relating to vehicles (e.g., the classes bus, car, and motorbike), however none of the images are taken from the dashcam of a car, which is a significantly different data distribution than photos taken with handheld cameras. For this reason, experimented with various 2D CNN backbones trained on the CityScapes [36] dataset for semantic segmentation, which is a large ego-centric driving dataset with pixel-level labels for 30 classes. The intuition is that this pre-trained backbone should result in better performance than using the PASCAL-VOC trained backbone.

The second architectural change was made to accommodate the types of labels found in the DOTA dataset. As mentioned previously, each anomalous frame in DOTA has both object detection labels and a frame-level anomaly classification label. The YOWO network however only predicts detections, and does not have the capability to predict frame-wise classifications. To solve this issue, we designed an additional anomaly classification head, which takes the network features as input (which are of shape batch x channels x 7 x 7) and globally pools the spatial dimensions to obtain a vector. This vector is then passed to a fully connected layer which then classifies the anomaly (i.e., its output feature is the same size as the number of anomalies, 11).

Loss Function. Our modified YOWO architecture outputs two predictions, a bounding box regres-

sion / classification and an anomaly classification. To this end, we use the original loss used in YOWO [31], L_{box}, with an added anomaly classification loss, L_{anom}. Our total loss function is:

$$L_{tot} = L_{box} + \lambda L_{anom}, \tag{1}$$

where L_{anom} is a standard multi-class cross entropy loss and λ is a weighting hyperparameter (see Sec 4.3 for ablations of this value).

4.3 RESULTS AND DISCUSSION

Implementation Details: We train all YOWO models with 16 frames at a sampling rate of 1 and a batch size of 48. During training, images were scaled to a size of 268 and then randomly cropped to a size of 224x224. During testing, images were rescaled to a size of 224x224. All models were trained for 200 epochs unless otherwise stated with a learning rate of 1e-3 which decays by a factor of 0.5 every 20 epochs. We used the ADAM optimizer with momentum of 0.9 and weight decay (L2) of 1e-4. All experiments were trained on the Vector Institutes computing cluster. All models were trained on four NVIDIA Tesla T4 GPUs and 32 CPUs, and took approximately 24 hours to complete.

Evaluation of 3D Backbones:

We first determine the best 3D backbone to use in the motion stream of the YOWO architecture. We compare four different backbones (listed by increasing number of parameters): ShuffleNet2-v2'[37], ResNet-18 [24], ResNet-50, and ResNext-101. The results from this experiment are shown in Fig. 14. Notable, the smallest model (ShuffleNet2-v2) achieves the best fscore. This could be because the dataset is on the small side, and therefore a network with lower capacity may overfit less to the training data. The anomaly classification accuracy was 39.4% for ShuffleNet2-v2 and 39.5% for 3D-ResNet18, however, we choose ShuffleNet2-v2 because the performance difference is greater when considering the fscore (see Fig. 14).

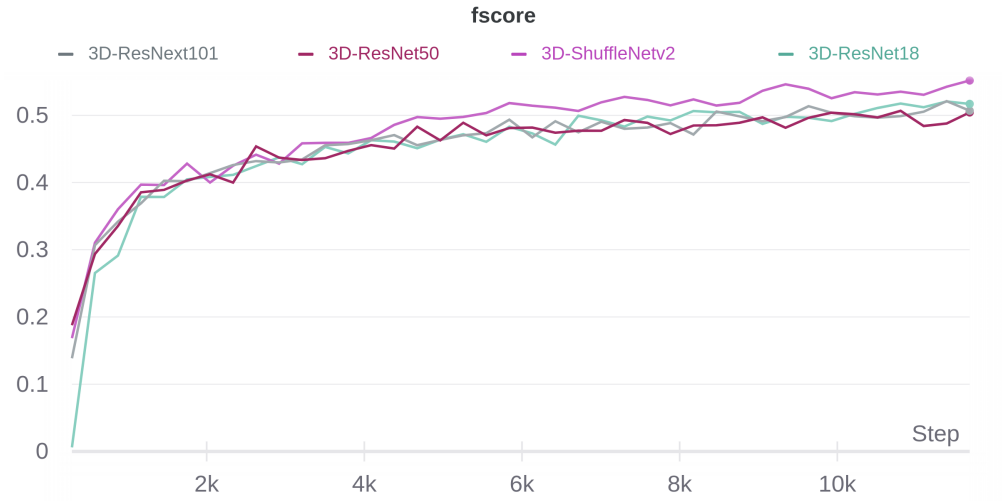


Figure 14: Experimenting with various 3D CNN backbones for YOWO. The smallest model, ShuffleNet2-v2, results in the best fscore.

Evaluation of 2D Backbones: In this experiment, we aim to see whether replacing the 2D back-

Class ID	1	2	3	4	5	6	7	8	9	Average
Anomaly Classification Accuracy	0.0	51.4	35.9	34.4	70.8	5.9	0.0	16.7	20.3	**48.3**

Table 5: Per-class anomaly classification accuracy on the DOTA validation set.

bones (i.e., appearance stream) with different architectures and/or pre-training strategies can benefit the YOWO model's final performance. Figure 15 shows the results, in terms of Fscore and anomaly classification accuracy, of YOWO with various 2D backbones. Here, 'Frozen' and 'Unfrozen' refer to the weights of the backbone being optimized during training or kept as the initialization weights. Note that all models are trained on the CityScapes [36] dataset, except for Darknet, which is trained on PASCAL VOC [33]. The results in Fig. 15 show that Darknet results in the best Fscore, while all models obtain similar performance in terms of the anomaly classification accuracy.

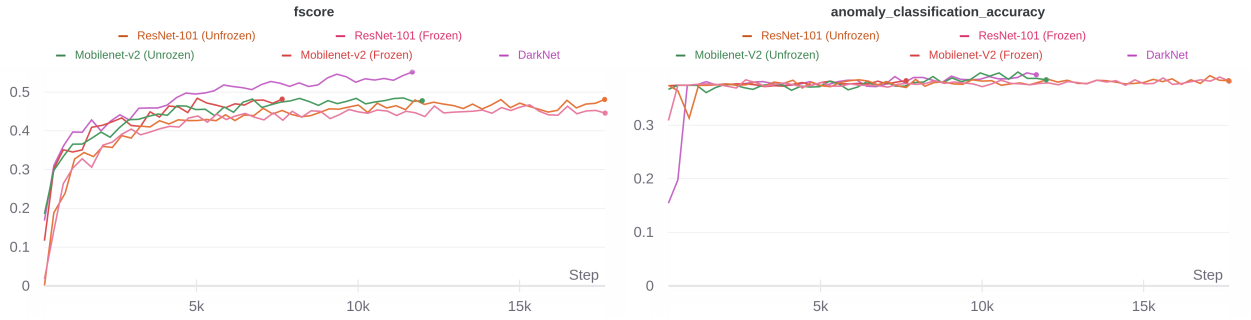


Figure 15: YOWO performance in terms of Fscore (left) and anomaly classification accuracy (right) with the appearance backbone replaced with various 2D CNN models. Darknet results in the best overall performance.

Anomaly Cross Entropy Weighting: The output of the model is penalized with a bounding box and classification loss for objects, and a cross entropy loss for the anomaly taking place in the video. Given that the losses are on different scales (e.g., the bounding box loss is between 10-300 while the cross entropy loss is between $0.5 - 4$) we place a weighting parameter, lambda, on the cross entropy loss. We run an ablation study with $\lambda = \{0.1, 0.5, 1, 5, 10\}$. We look at both the anomaly classification accuracy and bounding box loss to ensure that we are not sacrificing object detection performance for anomaly classification performance. From Fig. 16, we conclude that a lambda value of 5 gives the optimal trade-off between these two metrics, and so we use this value in all remaining experiments.

Final Model Performance: Given the results from the previous experiments, we now show the results for the final model. Table 2 shows the final per-class results in terms of anomaly classification accuracy. The anomaly classification accuracy for this model is 48.3% while the average object detection fscore achieved is 53.6%. These scores suggest that, while the model can correctly detect and classify anomalies about half the time, it is still premature for anything close to real-world deployment. We can also see the effect of the uneven data distribution in the class ID's

Qualitative examples of our final model are shown in Fig. 17. The model is capable of impressive detections in multiple weather conditions. Our model seems to perform well on common accident types. The most common occurring anomaly types involve being hit from the side and it can be

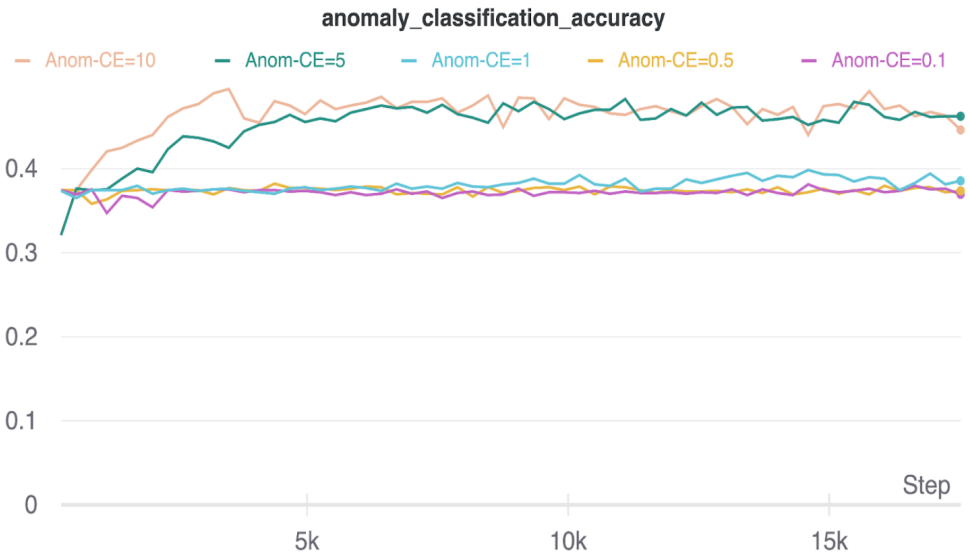


Figure 16: Ablating the anomaly classification weighting hyperparameter, λ. Our results show $\lambda = 5$ as the best candidate.

seen that our model often is successful for these anomalies (see Fig. 17, bottom-left) and gets an anomaly classification score on Class ID 3: 'Laterally Hit' of 35.9% and Class ID 5: 'Collision with Turning Vehicle' of 70.8%. However, there are specific scenarios where our model struggles. More specifically, it has difficulty detecting objects of interest that are far away from the driving vehicle, as well as more rare classes such as Class ID 6: ' Collision with Pedestrian' for which an accuracy of only 5.9% is obtained. Resolving these issues is left as a major focus for future work.

4.4 LIMITATIONS AND BEST PRACTICES

This project involved a number of technical challenges. One pervasive difficulty is that of overfitting, where the model performs well on the training set but poorly on the test set. A cause of this is the limited size of the DoTA dataset, which was further diminished due to about 2000 of the clips being removed from YouTube and therefore not accessible to the authors.

Overfitting issues were not overcome due to time and computational resource constraints. Limits on resources made it such that a full-fledged hyperparameter search could not be run, and the authors had to use judgement to determine a sequence of experiments and only tune a portion of the potential variables. Thus, there was no time left to experiment with backbones with less parameters and account for overfitting that way.

While the scope of this project was to determine the key subjects and type of anomaly conditional on there being an anomaly, one important step for future work is to determine which clips do contain anomalies and which do not. This would be beneficial for applications in which the user wishes to identify and focus on traffic accidents, and discard regular driving.

Figure 17: Qualitative examples of our final model. We show successful (left) and unsuccessful (right) predictions. The groundtruth and prediction are labelled in green and red, respectively.

5 IDENTIFYING CLINICALLY RELEVANT FEATURES OF INTEREST IN CHOLECYSTECTOMY PROCEDURES

Contributors: Kuldeep Panjwani, Shuja Khalid, Gabriel Chan, Vincent Zha

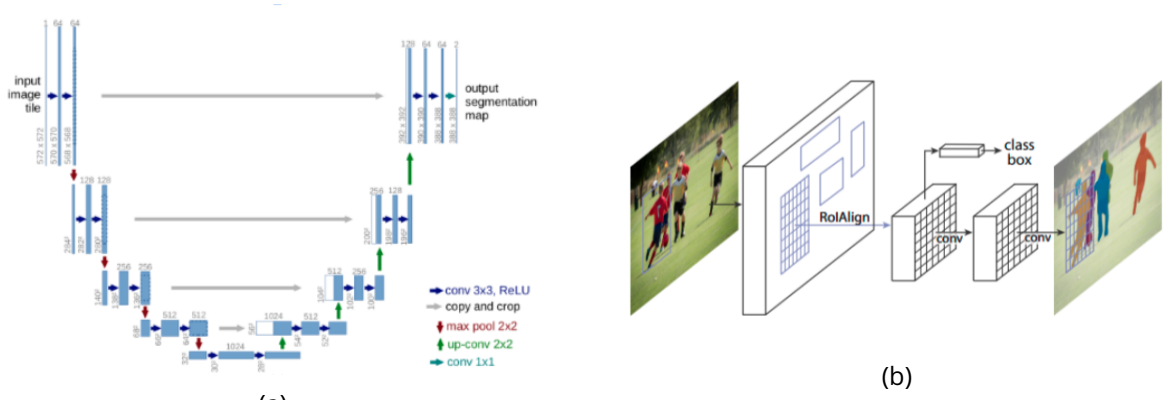
5.1 BACKGROUND/OBJECTIVE

Laparoscopic cholecystectomy (i.e., surgical removal of the gallbladder) is one of the most common surgeries performed in modern medicine (approximately 200,000/year in the US) [38]. Complications during difficult operations can result in longer recoveries, long-term disabilities, and even death. The goal of the operation is to remove the gallbladder and not injure other critical organs or structures that are in close proximity, such as the liver, intestine, bile ducts, arteries, and the portal vein. The goal of safe cholecystectomy has been based on the "critical view of safety (CVS)", which is defined by the complete and safe dissection of the cystic duct and artery at the base of the gallbladder. The goal of the project is to develop a near real-time tool for identifying critical regions during surgery to assist surgeons to arrive safely to the CVS.

5.2 METHOD

Model: Various computer vision methods such as object detection [39], semantic segmentation [17], and instance segmentation [40] may be used for this analysis. We decided that the most precise of the aforementioned approaches would capture the exact dimensions of the CVS (no-go zone) as opposed to an approximation of the intended region. Both semantic and instance segmentation techniques were explored for this purpose.

Semantic segmentation: Semantic segmentation is the process by which each pixel in an image



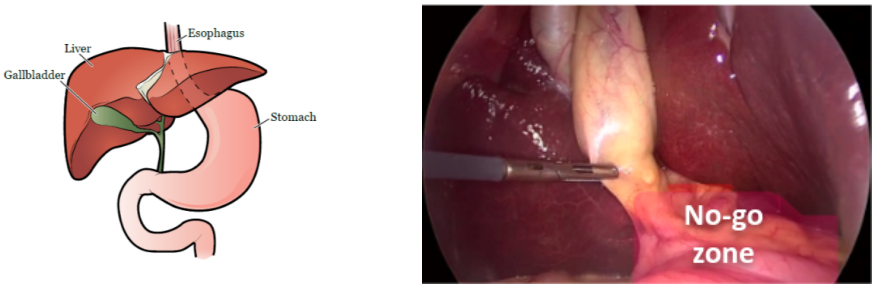
(a) (b)

Figure 18: UNet architecture [14] (a) and Mask-RCNN architecture [41] (b).

is labeled with a class. In semantic segmentation, the background pixels are also labeled, as this concept doesn't depend on the number of objects in an image but the class to which each pixel belongs [40]. A commonly used, supervised semantic segmentation model was trained using the provided annotations. The assigned classes are presented in Table 6. The assigned classes are presented in Table 6 and a high level architecture of the U-Net model is presented in Figure 18a.

Instance segmentation: Instance segmentation is based on both class recognition and instance recognition tasks. It goes beyond semantic segmentation in the sense that it can give more insight into the amount of objects belonging to a specific class within an image [40]. Instance segmentation allows for differentiating between different objects of the same class. A commonly used instance segmentation model, Mask-RCNN [41][5], is fine-tuned to the custom task. We employ a one-vs-all approach as we focus on only the pixels associated with the CVS. The assigned classes are presented in Table 6 and a high level architecture of the Mask-RCNN model is presented in Figure 18b.

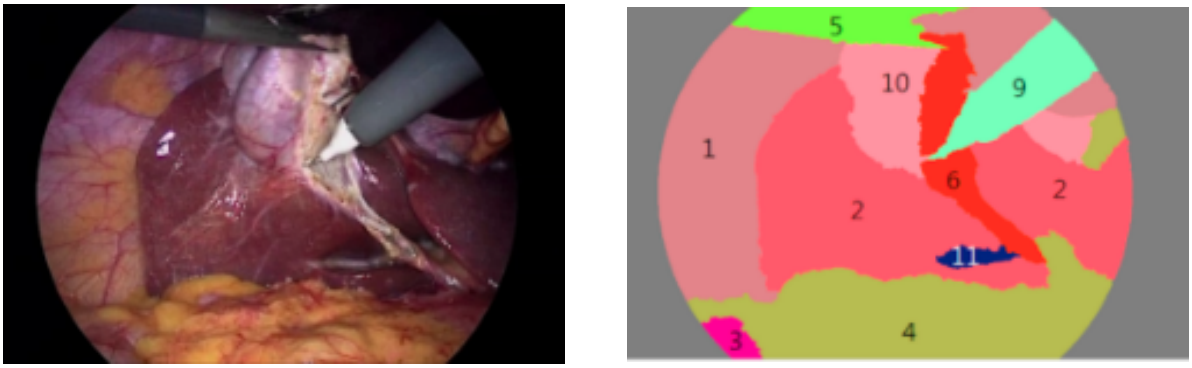
5.2.1 QUALITATIVE (FRAMES)



(a) (b)

Figure 19: An illustration of the relative position of the gallbladder [42] (a) and Illustration of the region of interest (b).

[5]https://github.com/facebookresearch/detectron2



(a) (b)

Figure 20: Input Image (a) and Target Output (b).

Table 6: Defined classes for the chosen modelling tasks.

Class ID	Class Name
1	Background
2	No-go-zone
3	Liver
4	Tools
5	Gallbladder
6	Fatty tissue

(a) Semantic segmentation

Class ID	Class Name
1	Background
2	No-go-zone

(b) Instance segmentation

Temporal smoothing: The Mask-RCNN model predictions are made at the frame level. While testing with video data, it became apparent that the resulting segmentations varied slightly across frames which manifested as flickering effects. To filter out this noise, we performed temporal smoothing by averaging the frame predictions across 5 frames throughout the video which resulted in less distracting videos. A more consistent approach and better smoothing algorithms will undoubtedly yield smoother outputs.

Dataset: There is a relative lack of datasets in the healthcare community, especially datasets consisting of surgical content. The Cholec80 dataset [43] is among the handful of publicly available datasets that has been used for providing proof-of-concept results for a variety of machine learning tasks in healthcare and surgery. The CholecSeg8k dataset consists of 8000 frames that have been annotated by Hong et al. [44]. A sample annotation is presented in Figure 20. Due to the challenging nature of the task, there are no baselines to compare against. It is also important to consider that the predicted classes presented in the paper have discernible bounds that may be used for prediction. The inclusion of an additional class (CVS) makes the modelling problem even more challenging.

Annotations For training the Mask-RCNN model, we collaborated with Dr. Gabriel Chan (*Assistant Professor of Clinical Medicine, Surgery - Hôpital Maisonneuve Rosemont*) to procure a list of annota-

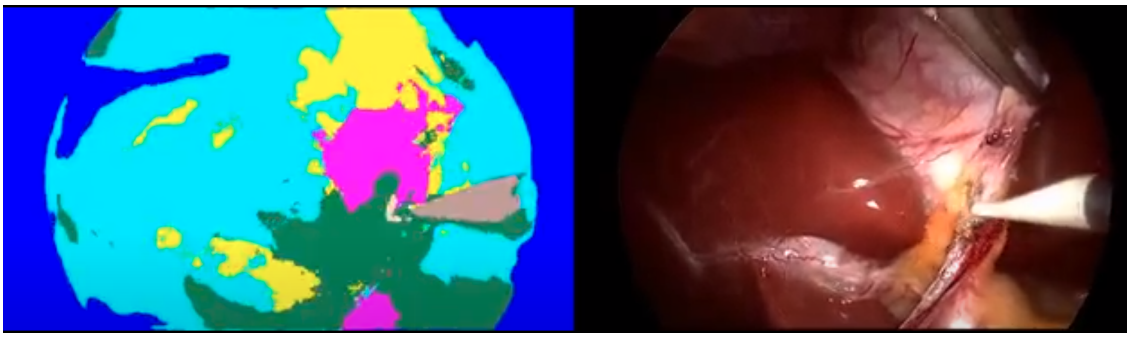


Figure 21: Sample output for the semantic segmentation task (left) using the *UNet* architecture.

Table 8: Results for the segmentation modelling task.

Feature	Dice Coefficient
No-go-zone	0.51
Liver	0.67
Tools	0.16
Gallbladder	0.50
Fatty tissue	0.71

tions for segmenting the CVS. An example segmentation is shown in Figure 19a and 19b. Over a period of three months, Dr. Chan and his team annotated approximately 1100 frames using the VIA annotation tool [45] which were used for training the Mask-RCNN model. It is important to note that not every frame consists of a no-go zone.

5.3 RESULTS AND DISCUSSION

5.3.1 SEMANTIC SEGMENTATION

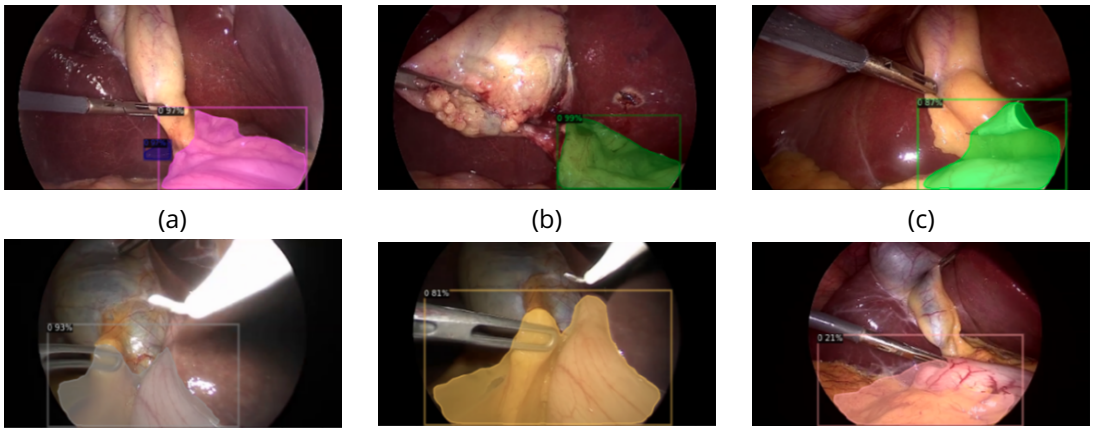
Figure 21 shows the qualitative results for the frame-wise segmentation where the per-pixel classifications are assigned. Without a holistic spatio-smoothing kernel, the resulting predictions are highly susceptible to specular effects as is seen in the results. The model is nevertheless able to correctly segment most of the important classes. The resulting quantitative results are presented in Table 8.

5.3.2 INSTANCE SEGMENTATION

After discussing the semantic segmentation results, it became apparent that not all of the classes presented in the semantic segmentation task would provide clinical value. The task was then slightly modified such that the same annotations were used but the task would focus on binary segmentation. Instead of assigning a per-pixel classification for every image, we chose to regress over a region and capture the set of pixels that would be described as the CVS. We use the Mask-RCNN model for this instance segmentation task. The qualitative results for instance segmentation are presented in Figure 22 and Figure 23. The instance segmentation task yielded an mAP of 0.51 for the no-go zone.

Table 9: Colour code for UNet multi-class segmentation

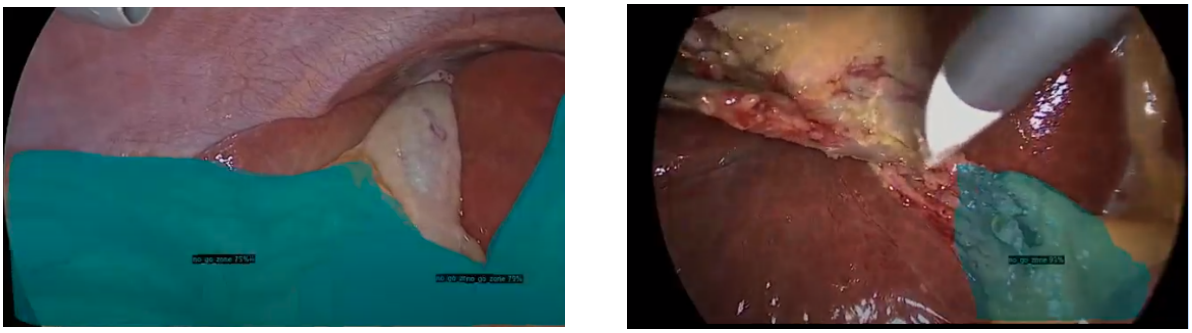
Feature	Colour
Liver	Blue
Tools	Reddish-gray
Gallbladder	Teal
Fatty tissue	Dark green



(a) (b) (c)

(d) (e) (f)

Figure 22: (a), (b) & (c) are acceptable segmentations as corroborated by our collaborator (Dr. G. Chan) whereas (d), (e) & (f) are potentially erroneous segmentations.



(a) Access (b) Dissection

Figure 23: Predicted no-go zone for during various steps of the surgical procedure, the model was only trained on annotated data from the dissection portion of surgery but is able to generalize across different procedures.

Annotations: Annotations were provided around the bounding box of the segment in case of the no-go zone area. The multi-class semantic segmentation colour code is in Table 9

5.4 LIMITATIONS AND BEST PRACTICES

Data: Surgical video data has many challenges pertaining to model training. One of the main challenges is due to varying lighting conditions. Videos captured through a laparoscopic camera are illuminated using an auxiliary light source for visibility [46]. Due to specular artifacts resulting from the outer membrane of internal organs, this illumination source can saturate pixels leading to inconsistent segmentation results. Another challenge encountered was due to the gelatinous and deformable texture of organs. During each surgery, organs are grabbed by the grasper. This causes a change in the shape of each region to be labeled within the frame, which is another source of uncertainty for the model's object and class detection functionalities [47].

Quantity of annotations required: Surgical data is manually labeled by qualified annotators within the medical field. This causes a potential lack of annotators and, therefore, smaller datasets on which to train models. Ideally, most of the frames in the video dataset would be training samples and only about 25% would be left for testing and validation. This would help ensure a dataset with a more realistic data distribution of the no-go zones for the model to learn. However, this might not currently be practical as the videos run at 25 frames per second and the video length can be around 30 minutes long. This leaves the medically qualified annotators to manually annotate approximately 30,000 frames per laparoscopic video. One of the methods used to combat this issue is data augmentation. However, data augmentation also has its limitations as it can only generate a limited amount of variations using image transforms [47].

Consistency of annotations/lack of quality control: The no-go zone is a region consisting of multiple parts of the digestive system being operated on. Therefore, it is prone to annotation inconsistencies given individual frames from a surgical video, as the annotators may manually label certain segments which are not technically part of this region. One of the main occurrences of this case is where the grasper overlaps the no-go zone. In many cases, the annotators manually annotated the no-go zone on top of the grasper to indicate that the no-go zone was under it. Although this is an understandable error, it likely caused a drop in performance as the model potentially linked the tool itself with the no-go zone.

Prone to overfitting due to the similar frames being annotated: There were approximately 1000 annotated frames that were available for training and validation. Therefore, most of these frames were very similar in nature while only having certain arrangements of surgical tools and camera angles. By training our model over several epochs with this limited sample size, it will learn to handle only a given amount of cases. Some of the techniques to deal with this issue is to tune hyperparameters such as implementing a decaying learning rate, increasing regularization, and performing data augmentation [47].

5.5 CONCLUSION AND FUTURE DIRECTIONS

The current model serves as a proof-of-concept that yields good results despite the lack of distinct landmarks. The model should generalize to different capture platforms as different hospitals use different systems. The training data should evenly sample cases where there is a large amount of bleeding and re-train using these cases. Such a scenario is common during surgery and would further enforce the need for an augmented system by providing surgeons with the location of the no-go zone in more challenging surgeries. Such a model would provide real-time feedback to sur-

geons such that they would receive real-time prompts. The quality of segmentation annotations can significantly vary if there isn't a proper framework provided to analysts. Having a system in place to verify and quality check annotations would yield more improved and consistent results. More analysis is also required to determine the relative trade-offs between coarse and fine annotations.

6 TRANSFER LEARNING FOR EFFICIENT VIDEO CLASSIFICATION AND DETECTION

Contributors: Raghav Goyal, Xin Li, Andriy Levitskyy

6.1 BACKGROUND/OBJECTIVE

The task of Spatio-temporal action detection involves classifying and localizing actions both in 2-dimensional space (*x* and *y*) and 1-dimensional time coordinates. The annotations required for this task are bounding boxes which localize actions in space (e.g. boxes around actors performing actions) and tracks actions over time, effectively forming a 3D volume also known as action tubelet [48]. Gathering such annotations can be prohibitively expensive as it not only annotating bounding boxes in space but also in time. Our aim in this project is to study this task in the light of few or zero annotations. In particular, we use multi-modal similarity measures (lingual and visual) to transfer concepts from an existing, readily-available video classification database to the task of spatio-temporal action detection. The multi-modal similarity measures require zero to a few target annotations ($< 5\%$), and we provide benchmarks and analysis of the results.

6.2 RELATED WORK

Learning with few annotations has been explored in the task of object detection. In particular, the approaches in this category assume two sets of classes: *base* and *novel*, where *base* classes have abundant data and *novel* classes have few or none. The overall idea is to transfer concepts from *base* to *novel* classes using a similarity measure between the two sets of classes, in effect forming object detectors for *novel* classes by leveraging already available detectors of *base* classes [49, 50]. Another set of closely related works use object and scene priors to contextualize actions for the task of action detection [51, 52]. A common theme among these works is to form priors for an action based on scenes in which they occur, objects present while performing the action, and also other spatial and semantic priors to help classify an action. Notably, Mettes *et al.* [52] uses such priors to draw up detectors for the task of Spatio-temporal action detection which requires no data and provide performance on AVA dataset [53]. In this work, we approach the task of Spatio-temporal action detection within the framework of transfer learning, and we use existing action classifiers trained on a set of classes of an existing database, and transfer the classifiers to our target classes using a similarity measure between the classes.

6.3 METHOD

Dataset: We used the atomic visual actions (AVA) dataset [53] which consists of 80 action classes on 15-min video clips taken from movies. For every annotated frame in a video, all the actors are

localized using bounding boxes along with the action label. Such annotations form the Spatio-temporal action annotations. We used AVA v2.2 The train set consists of (provide stats).

Approach Overview: We transferred off-the-shelf, pre-trained action classifiers to the task of spatio-temporal action detection. In particular, we used action classifiers already trained on the Kinetics dataset [54], which contains roughly half-a-million 10-sec clips over 600 action classes. Since the task of action classification does not require localization, the action classifiers we obtain takes as input a video clip and provide predictions over 600 Kinetics classes. The action classifiers are composed of a feature extractor - which takes a video and maps it to intermediate features, followed by a spatio-temporal pooling layer and a fully-connected layer to map those features to output predictions over 600 classes. For the task of spatio-temporal action detection on the AVA dataset, we are provided with bounding boxes of persons performing actions in a video, so we do not need to localize actors. In this case, a feature extractor takes a video clip, extracts features and uses the provided person bounding boxes to sample features specific to each person using temporal pooling and RoIAlign [55] operations, and finally uses a fully-connected layer to predict over 80 actions.

Model specifications: We used SlowFast [56] model as a feature extractor and make use of their code [57] to form our experiments. Specific to the SlowFast model architecture, we observed that the feature extractor for both AVA and Kinetics action classification uses the same backbone, and differs only in the pooling layers to generating features for final classification. Specifically, with reference to Fig 24, the Kinetics' model architecture pools features across a whole video clip for action classification, while AVA's model architecture pools features specific to each person using RoIAlign for final classification.

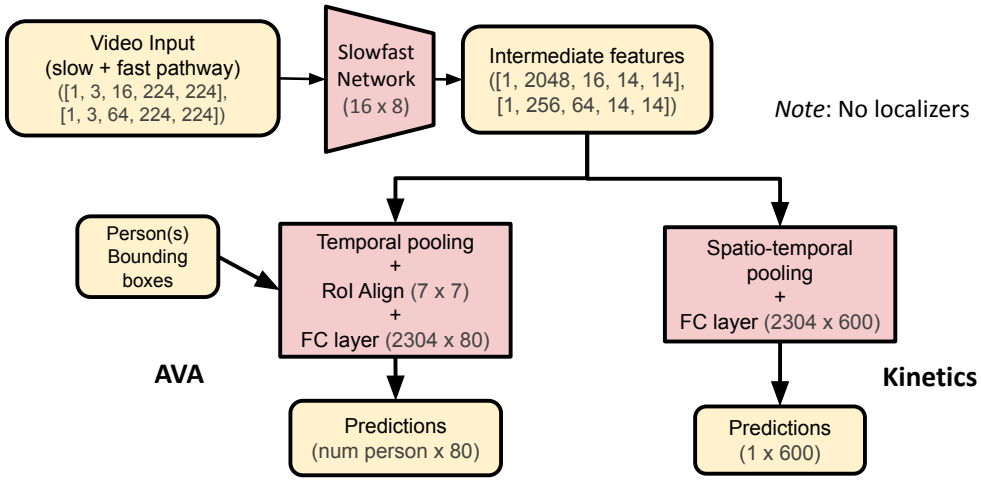


Figure 24: **Difference between AVA and Kinetics model architecture.** Specific to SlowFast [56] model architecture, the Kinetics and AVA pipeline differ in how the pooling is done. In Kinetics, pooling is done across the whole video, while in AVA, pooling is done according to each person bounding box provided.

6.3.1 TRANSFER MECHANISM

We used the above observation to come up with a transfer scheme from Kinetics to AVA which does not involve any extra parameters. Since the only difference between the two architectures is the

use of non-parameterized pooling and RoI Align layers, we can replace the fully-connected layer in AVA model with the fully-connected of Kinetics model and effectively obtain Kinetics prediction for each person in AVA dataset. This is shown in Fig 24. In summary, we used pretrained Kinetics model to obtain Kinetics predictions for each person in AVA dataset without losing generality or using any extra parameters. The remaining part is to map Kinetics predictions over 600 classes to 80 AVA classes using a similarity function $\mathbf{S} \in \mathbb{R}^{80} \times \mathbb{R}^{600}$ which we will describe further.

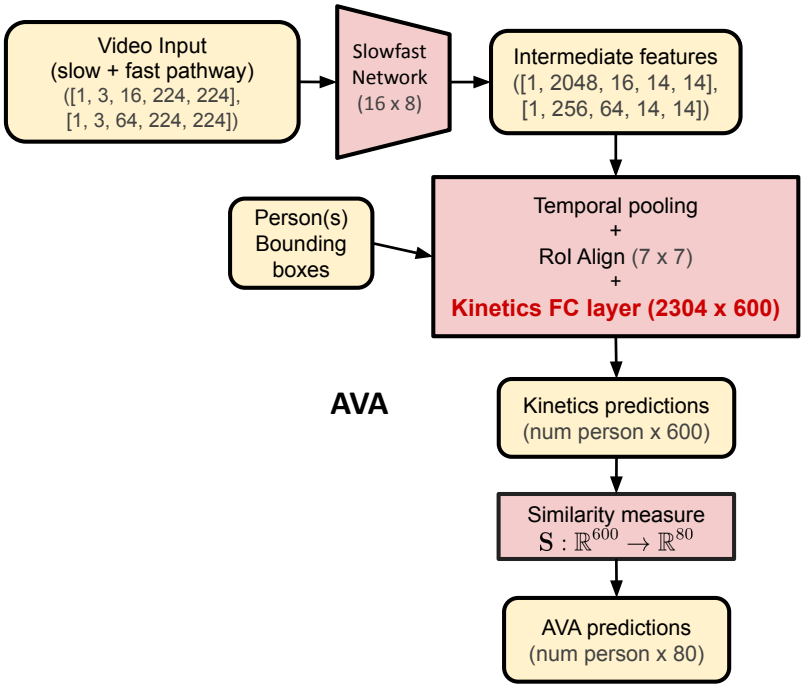


Figure 25: **Kinetics to AVA transfer.** The figure illustrates that we first obtain Kinetics predictions for every person bounding box using off-the-shelf Kinetics classifiers. Then we use a similarity measure $\mathbf{S}$ which maps the Kinetics predictions to AVA predictions.

Lingual similarity. We used an off-the-shelf language model pretrained on web-sourced data[6] (blogs, news, comments) to embed class labels into 300-dimensional vectors. After obtaining embedding for both AVA and Kinetics class labels, we computed cosine similarities between the two sets of AVA and Kinetics class labels to obtain a matrix $\mathbf{S_{lin}} \in \mathbb{R}^{80} \times \mathbb{R}^{600}$.

Visual similarity. We sampled 5% of the annotated data (or videos) uniformly from the train set, thereby maintaining the same distribution of number of samples per label as the train set. We obtained Kinetics predictions (as described in Fig 25) for every annotated bounding box in the sampled data. Since we are given ground-truth AVA label for every annotation, we grouped the Kinetics predictions based on ground-truth AVA label. Essentially, for the 5% sampled data, we ran Kinetics classifier for each labelled bounding box, and based on the ground-truth AVA label we group the Kinetics predictions into a list.

After obtaining a list of Kinetics predictions for every AVA class label, we averaged out the list to obtain a 600-dimensional vector for each AVA class label, thereby forming our visual similarity

[6] https://spacy.io/models/en#en_core_web_md

Table 10: Performance of different strategies and similarity measures.

Strategy	Performance (mAP)
Random	3.9
Visual similarity	5.9
Lingual simple averaging	6.7
Lingual weighted averaging	7.2
Fully-supervised	29.1

$\mathbf{S_{vis}} \in \mathbb{R}^{80} \times \mathbb{R}^{600}$. We can interpret the similarity in the following way, for every AVA annotation, we assume that Kinetics classifiers capture the semantics of the actions, for e.g., annotations of an AVA class `fight` would on average expect the Kinetics classifiers to predict semantically similar classes conveying aggression (such as `sword fighting`) compared to other unrelated classes (such as `brushing teeth` or `crawling baby`). Such associations or regularities would be reflected in visual similarity $\mathbf{S_{vis}}$.

6.4 RESULTS AND DISCUSSION

Implementation Details: We used four Nvidia Tesla-T4 GPUs for generating visual similarity from 5% of the train set, and for evaluating our model with on validation set.

Strategies for generating classifiers: Since using all similarity values can be noisy, we experimented with different strategies. The **random** strategy produces random predictions over AVA classes. The **lingual simple averaging** uses top three most similar Kinetics classes for every AVA class and average out their predictions. With respect to Figure 25, this would be similar to taking each row of $\mathbf{S_{lin}}$, picking top three values and substituting them with $\frac{1}{3}$, and zeroing out rest of the values. **lingual weighted averaging** strategy is similar but it takes a weighted sum of predictions instead of averaging them, where weights are similarity values. The **visual similarity** strategy is similar to the lingual counterparts but it uses visual similarity $\mathbf{S_{vis}}$.

6.4.1 QUANTITATIVE RESULTS

The performance of different strategies can be found in Table 10. As expected, **random** strategy performs worst which indicates that our strategies despite using little to no training data makes meaningful predictions. We also observe that **lingual similarity** performs better than **visual similarity**. And that the weighted version of **lingual similarity** performs better than simple averaging which indicates that relative differences among similarity values contains meaningful semantic information.

6.4.2 QUALITATIVE RESULTS

Lingual similarity. We look at class-wise performance comparisons between random and lingual strategy, and list top five positive and negative relative differences in Table 11. We observe that AVA classes which benefit the most have semantically close classes in Kinetics dataset. For e.g. the top three similar Kinetics classes to the AVA class `swim` are different techniques of `swim` which provides significant performance boost. On the other hand, AVA classes which degrade in performance have

Table 11: Top Positive and Negative relative performance differences using Lingual similarity.

AVA class	Performance Random Strategy (x)	Performance Lingual similarity (y)	Relative difference ((y - x) / x)	Top-3 similar Kinetics classes and their similarity values
swim	0.02	26.1	964.8	swimming front crawl (0.7038) swimming backstroke (0.6811) swimming butterfly stroke (0.6676)
dance	0.02	5.9	203.7	tango dancing (0.9695) dancing ballet (0.8650) jumpstyle dancing (0.8347)
walk	0.2	26.6	136.6	walking the dog (0.7154) walking through snow (0.7121) moon walking (0.7041)
open (e.g. a window, a car door)	0.1	5.7	113.3	opening door (0.6833) opening present (0.6243) push up (0.5620)
enter	1.4	0.06	-0.955	waiting in line (0.4947) opening present (0.4914) giving or receiving award (0.4648)
crouch/kneel	42.0	2.0	-0.953	massaging legs (0.7172) bending back (0.7164) shaking hands (0.6877)
bend/bow (at the waist)	29.4	2.5	-0.914	tying bow tie (0.7165) bending back (0.6681) bending metal (0.5856)
lift (a person)	0.83	0.22	-0.733	snatch weight lifting (0.6355) lifting hat (0.5926) pushing wheelchair (0.5650)

less semantic matches (lingually) in Kinetics dataset. For e.g. the class `enter` has no direct match in the Kinetics dataset. We also observe that lingual similarity can also produce unintended similar classes. For example, in the case of class `bend/bow`, the most similar class is `tying bow tie` which makes sense linguistically by the use of word "bow" but differs semantically.

For computing lingual similarity we found preprocessing of class names to be crucial, as a few AVA class names contain expanded description in parenthesis which produces non-meaningful similarities. For example, the AVA class `kiss (a person)` tries to linguistically find similar class names to `kiss` and `person` which fails to associate a direct match `kissing` in the Kinetics dataset. Therefore, getting rid of the parenthesis terms improved the performance.

Visual similarity. Top positive and negative associations are shown in Table 12. We note that there are positive associations such as the AVA class `answer phone` is able to find it's counterpart in Kinetics dataset `talking on cell phone` visually. However, we note for few classes it doesn't work, such as `walk`, `swim`.

We attribute the negative visual associations to two reasons, (1) **domain mismatch**: the data domains of AVA and Kinetics are not the same since AVA is annotated on top of movie clips while Kinetics is sourced from YouTube. This creates a domain mismatch and results in classes which do

Table 12: Positive and negative associations in Visual similarity. The table shows positive and negative associations for AVA classes along with their top similar Kinetics classes.

Positive associations		**Negative associations**	
AVA class	**Top similar Kinetics classes**	**AVA class**	**Top similar Kinetics classes**
dance	celebrating (0.10) salsa dancing (0.07)	walk	opening door (0.06) acting in play (0.04)
lie/sleep	sleeping (0.14) crying (0.06)	close	opening door (0.36) opening refrigerator (0.08)
answer phone	talking on cell phone (0.39) crying(0.07)	brush teeth	air drumming (0.5) tasting food (0.49)
push	pushing wheelchair (0.13) pushing car (0.06)	swim	jumping jacks (0.42) waiting in line (0.15)

Table 13: Qualitative differences between Lingual and Visual similarities.

Lingual similarity	**Visual similarity**
Data independent (Requires no target data to form similarity)	Data dependent (Requires some target data to form similarity)
Less adaptive to data	Has potential to be more adaptive to data
Less susceptible to noise in data	More susceptible to noise in data

not necessarily have semantically similar matches in both the datasets. Such as the AVA class `close` is visually close `opening door` but doesn't have a direct corresponding match in Kinetics dataset because it has a too meaning, (2) **class imbalance**: since the AVA dataset long-tailed, forming visual similarity using the data distribution leads to imperfect visual matches for classes lying on long tails. For e.g. in the 5% subsampled version of AVA dataset, the class `brush teeth` is present only 2 times while the class `watch` is present $44,471$ times. Such imbalance affects the visual associations negatively.

6.5 LIMITATIONS AND BEST PRACTICES

Overall, we studied the multi-modal similarity measures to transfer knowledge from a readily-available source dataset to a target dataset which is expensive to annotate. In the process we use few to no target annotations and found our approach to yield better results than random strategy. We observe qualitative and quantitative differences between lingual and visual similarities, and we sum up the differences in Table 13.

Notably, visual similarity is dependent on data which works in its favour that it can adapt to data but also leaves it susceptible to noise in data (such as class imbalance). On the other hand, lingual similarity can form similarities using only the class names and requires no target data which could be desirable in situations where no target data is available.

7 CONCLUSION AND FUTURE DIRECTION

Throughout the project, participants gained exposure to a variety of computer vision techniques and use cases, allowing them explore ways in which computer vision tasks could be applied to real-world applications. Beyond the project, there are a number of interesting avenues that can be explored.

In terms of general approaches for future applications, some self-supervised learning models have been found to perform better than supervised learning in certain object detection and semantic segmentation tasks. [58]

An additional trend that can potentially mitigate data volume concerns is auto-annotation, which may be especially helpful in clinical settings where use cases are narrow and data is sparse.

In terms of anomaly and semantic segmentation application in image data, potential future directions include using techniques such as open-set segmentation, which effectively combine both anomaly detection and semantic segmentation to identify both predefined classes and anomaly classes, as well as applying the models to different data sets to evaluate performance and determine additional benchmarks.

In terms of semantic segmentation applications in cholecystectomy videos, one major challenge was the limited amount of labelled data available. Future mitigation approaches include further hyper-parameter tuning, increasing regularization and using data augmentation techniques to increase the volume of training data.

In terms of the video applications in traffic incident detection, future work could focus on different, larger data sets that prevent over-fitting. A related direction is to conduct additional hyper-parameter tuning and experiment with backbones to account for potential over-fitting issues.

8 REFERENCES

[1] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018.

[2] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[3] Hayden Gunraj, Linda Wang, and Alexander Wong. Covidnet-ct: A tailored deep convolutional neural network design for detection of covid-19 cases from chest ct images. *Frontiers in medicine*, 7, 2020.

[4] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[5] Christoph Baur, Stefan Denner, Benedikt Wiestler, Nassir Navab, and Shadi Albarqouni. Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study. *Medical Image Analysis*, page 101952, 2021.

[6] Sina Akbarian, Tania Cawston, Laurent Moreno, Samir Patel, Vanessa Allen, and Elham Dolatabadi. A computer vision approach to combat lyme disease. *arXiv preprint arXiv:2009.11931*, 2020.

[7] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019.

[8] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[9] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[11] Antoni Chan and Nuno Vasconcelos. Ucsd pedestrian dataset. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(5):909–926, 2008.

[12] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[13] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Jeremy Jordan. An overview of semantic image segmentation. `https://www.jeremyjordan.me/semantic-segmentation/`, 2020.

[16] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[18] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[20] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018.

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

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1099–1106. IEEE, 2016.

[26] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018.

[27] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2152–2161, 2019.

[28] Toshiaki Ohgushi, Kenji Horiguchi, and Masao Yamanaka. Road obstacle detection method based on an autoencoder with semantic segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[29] Jeremy Collins. Rise of dashcams & their impact on safety. `https://www.azuga.com/blog/rise-of-dashcams-their-impact-on-safety`. Accessed: 2021-11-23.

[30] Yu Yao, Xizi Wang, Mingze Xu, Zelin Pu, Ella Atkins, and David Crandall. When, where, and what? a new dataset for anomaly detection in driving videos. *arXiv preprint arXiv:2004.03044*, 2020.

[31] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019.

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

[33] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, jun 2010.

[34] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[35] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[36] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[37] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[38] Esteban Varela and L Michael Brunt. Sages laparoscopic surgery safety checklist. In *The SAGES Manual of Quality, Outcomes and Patient Safety*, pages 77–84. Springer, 2012.

[39] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[40] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[41] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[42] Memorial Sloan Kettering Cancer Center. About your gallbladder removal surgery. `https://www.mskcc.org/cancer-care/patient-education/about-your-gallbladder-removal-surgery`. Updated: 2013-01-14, Accessed: 2021-11-20.

[43] Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Endonet: a deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2016.

[44] W-Y Hong, C-L Kao, Y-H Kuo, J-R Wang, W-L Chang, and C-S Shih. Cholecseg8k: A semantic segmentation dataset for laparoscopic cholecystectomy based on cholec80. *arXiv preprint arXiv:2012.12453*, 2020.

[45] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. In *Proceedings of the 27th ACM international conference on multimedia*, pages 2276–2279, 2019.

[46] WebMD. Laparoscopic surgery: Purpose, procedure, and benefits. `https://www.webmd.com/digestive-disorders/laparoscopic-surgery`. Published: 2021-03-06, Accessed: 2021-11-20.

[47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[48] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4405–4413, 2017.

[49] Krishna Kumar Singh, Santosh Divvala, Ali Farhadi, and Yong Jae Lee. Dock: Detecting objects by transferring common-sense knowledge. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 492–508, 2018.

[50] Siddhesh Khandelwal, Raghav Goyal, and Leonid Sigal. Unit: Unified knowledge transfer for any-shot object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5951–5961, 2021.

[51] Zuxuan Wu, Yanwei Fu, Yu-Gang Jiang, and Leonid Sigal. Harnessing object and scene semantics for large-scale video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3112–3121, 2016.

[52] Pascal Mettes, William Thong, and Cees GM Snoek. Object priors for classifying and localizing unseen actions. *International Journal of Computer Vision*, 129(6):1954–1971, 2021.

[53] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[54] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zissernman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[55] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[56] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[57] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. `https://github.com/facebookresearch/slowfast`, 2020.

[58] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin1, and Piotr Bojanowski1. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988v2*, 2021.