A Comparison of the Performance of 2D and 3D Convolutional Neural Networks for Subsea Survey Video Classification

Anastasios Stamoulakatos*, Javier Cardona*, Craig Michie*, Ivan Andonovic*, Pavlos Lazaridis†, Xavier Bellekens*, Robert Atkinson*, Md. Moinul Hossain‡, Christos Tachtatzis*
*Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, UK
†Department of Engineering and Technology School of Computing and Engineering, Huddersfield, UK
‡School of Engineering and Digital Arts, University of Kent, Canterbury, Kent, UK

***Abstract*—Utilising deep learning image classification to automatically annotate subsea pipeline video surveys can facilitate the tedious and labour-intensive process, resulting in significant time and cost savings. However, the classification of events on subsea survey videos (frame sequences) by models trained on individual frames have been proven to vary, leading to inaccuracies. The paper extends previous work on the automatic annotation of individual subsea survey frames by comparing the performance of 2D and 3D Convolutional Neural Networks (CNNs) in classifying frame sequences. The study explores the classification of burial, exposure, free span, field joint, and anode events. Sampling and regularization techniques are designed to address the challenges of an underwater inspection video dataset owing to the environment. Results show that a 2D CNN with rolling average can outperform a 3D CNN, achieving an Exact Match Ratio of 85% and F1-Score of 90%, whilst being more computationally efficient.**

***Index Terms*—Deep Learning, Subsea Inspection, Video Classification, Underwater Pipelines**

I. Introduction

SUBSEA pipeline inspection is an essential process within the Oil and Gas industry to ensure interrupted production as any potential risk must be identified prior to criticality in order to mitigate equipment damage and environmental threats. Remotely Operated Vehicles (ROVs) are employed routinely for subsea pipeline and power transmission cable inspections [1]. These vehicles are submerged and driven above the pipeline, controlled via a cable connected to an offshore vessel, acquiring inspection data from various sensors (e.g., visual, echo-sounders, laser scanning and magnetometers sensors) [2]. The data are inspected by survey supervisors on the vessel to assess the overall condition of a key asset. Following the capture of the inspection video, data coordinators record logs of key events observed during the survey both in real-time and subsequently through a Quality Control (QC) process. Timestamp and location are used to annotate events such as pipeline exposure, burial, field joints, anodes, free spans, and debris; examples of such events are illustrated in Figure 1. Given the importance of subsea pipeline inspection, coupled to the challenges of capturing subsea visual footage e.g., highly variable illumination and sea life, marine growth, sand algae [3], a range of methods to automate the annotation process have been proposed in recent years [4]–[7], to reduce the annotation burden, increase reliability, robustness and inspection speed.

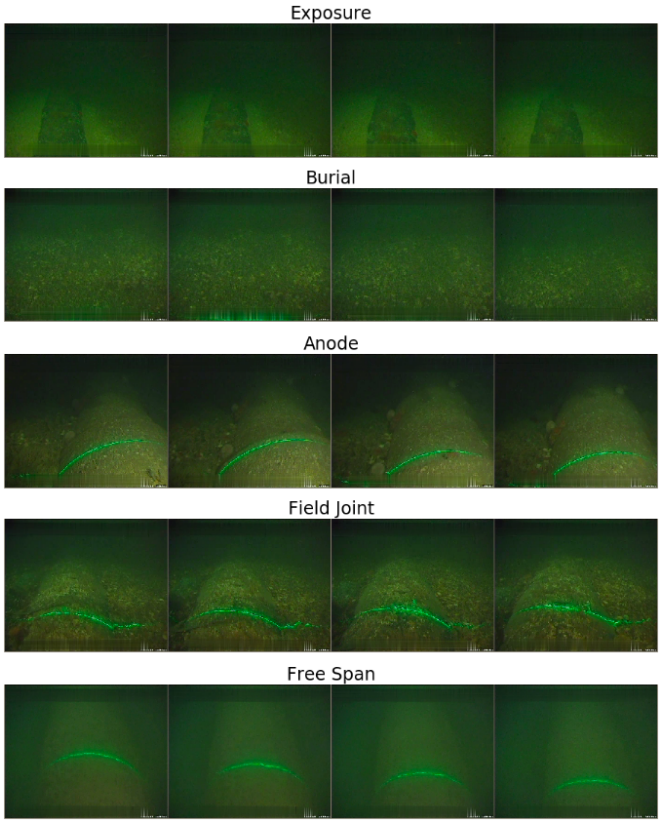

Fig. 1. 4-frame Examples of Events of Interest

Here, previous research for the annotation of subsea pipeline surveys [7] is extended to video frame sequence classification. The main contributions are summarized as follows:

The work was partially supported by The Data Lab Innovation Centre, Edinburgh, Scotland, UK (project registration code 16270), the Oil and Gas Innovation Centre, Aberdeen, Scotland UK (project registration code 18PR-16) and N-Sea, Zierikzee, Netherlands.

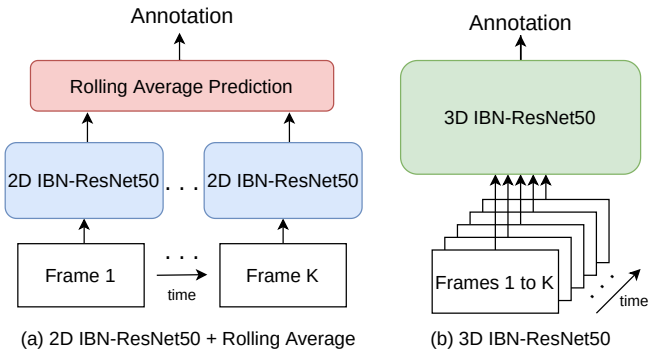

Fig. 2. 2D and 3D Spatio-Temporal Architectures for Annotation of Subsea Pipeline Surveys

- The use of an extended dataset, comprising short video clips for the five events of interest (exposure, burial, free span, field joint, and anode), instead of individual frames. The nature of the dataset is such that high class imbalance and label noise are present inherently. A balanced sampling and label smoothing techniques have therefore been implemented, in addition to spatial data augmentation, to make the model invariant to camera positioning/movement and improve generalisation.
- A 2D IBN-ResNet50 [8] network that classifies individual frames followed by a Rolling Average mechanism is compared to a 3D IBN-ResNet50 architecture that creates a single prediction per video clip. Both network architectures utilise Instance Batch Normalisation (IBN) [9] between convolutional layers to improve model performance for data acquired under varying lighting conditions and colour contrast changes. The network architecture for both models are shown in Figure 2.

II. Related Work

The automation of Subsea Pipeline Inspection relies on data collected from the instrumentation mounted on Remote Operating Vehicles. Jacobi *et al.* [10], [11] proposed a pipeline tracking method for Autonomous Underwater Vehicle (AUV) guidance by fusing optical, magnetic, and acoustic sensor data collected in a simulation setting. Bharti *et al.* [12] used multi-beam echo-sounder data for pipeline detection and estimation of orientation. Narimani *et al* [13] proposed a pipeline and cable tracking technique that determines the angle between the vehicle and pipeline by converting the images to grey-scale and applying the Hough transformation. A real-time vision-based detection system for underwater pipelines using edge-based image processing to detect pipeline contours and a Kalman filter for de-noising was developed by Zingaretti *et al.* [14], [15]. Ortiz *et al.* [16] proposed a method for identifying susbsea cable contours in tandem with a linear Kalman filter to predict the contours in the following frame whilst Asif *et al.* [17] proposed a pipeline tracking method that utilises the Bresenham line algorithm and B-Spline to detect noise-free pipeline contours. Khan *et al.* [18] reported on a method for underwater pipeline image enhancement using wavelet auto-encoding and K-means for clustering corrosion segments. All reported approaches to date utilise traditional signal processing on the sensor data to detect and track pipeline contours, unlike the work presented here which classifies specific events of interest.

Recently, the adoption of deep learning approaches to process subsea imagery has yielded excellent performance in multiple underwater applications. Martin-Abadal *et al.* [19] proposed a fully convolutional network for the semantic segmentation of Posidonia Oceanica and deployed it on a Turbot AUV for online segmentation of meadows. Obyrne *et al.* [20] utilises photo-realistic synthetic imagery for training SegNet [21] for bio-fouling detection on marine structures. A deep learning method for classifying coral reefs, trained on images of the sea floor acquired by ROVs and AUVs was proposed by Mahmood *et al.* [22] whilst Jeon *et al.* [23] introduced a pose estimation network for underwater objects, relying the utilization of synthetic data to improve underwater deep learning approaches. King *et al.* [24] carried out a comparison between four Fully Convolutional Neural Networks (FCNN) [25] for semantic segmentation of coral reef images; Fulton *et al.* [26] developed a deep learning approach for detecting trash in realistic underwater environments; Xu *et al.* [27] used YOLO [28] for fish detection in real-world water power sites; and Guo *et al.* [3] proposed a Generative Adversarial Network [29] that enhances the quality of underwater images.

A number of deep learning based methods have been proposed for subsea pipeline inspection and event annotation applications. Petraglia *et al.* [5] compared a Multilayer Perceptron (MLP) with a single hidden layer, trained on features extracted from 3-level Wavelet decomposition, with a Convolutional Neural Network (CNN) classifying four types of events: inner coating exposure, algae, flange and concrete blankets. The CNN outperformed the MLP without the need for manual feature extraction. Bharti *et al.* [6] fine-tuned U-Net [30] in a self-supervised setting utilising multi-beam echosounder data for detection and segmentation of subsea pipelines. In our previous work [7], a framework for automatic subsea pipeline image annotation was presented by using transfer learning on ResNet-50 [31] in a multi-label image classification setting with Precision-Recall curves for optimal threshold selection.

Deep learning models presented in the literature have shown good performance on single frames. The study reported here details a comparison of two CNN-based spatio-temporal architectures designed to automate subsea survey video annotation. The two models are compared based on their classification performance on survey frame sequences, training and inference times as well as the samples used in their training, representing the first reported use of deep learning approaches that operate directly on subsea pipeline video data instead of single frames.

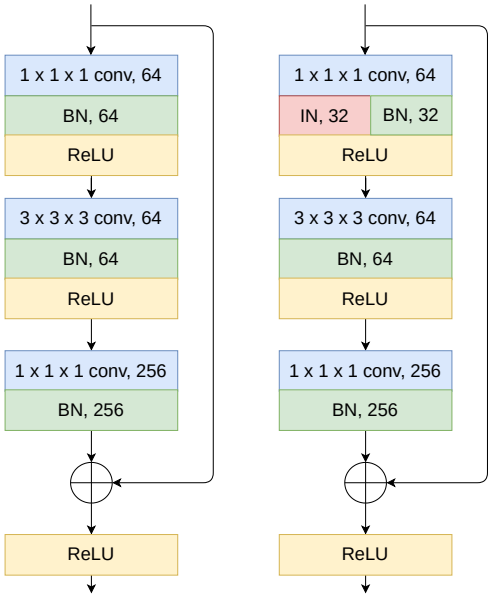

Fig. 3. 3D ResNet block and 3D IBN-ResNet block

III. Model Architecture

The core task is to classify/annotate images or frame sequences of subsea pipelines. CNNs have properties such as local connectivity, shared weights and spatial downsampling that make them ideally suited to this task [32]. Although CNNs have been applied extensively in the computer vision domain, they have also been utilised in the analyses of various data in multidimensional arrays. For example, a 1D CNN can be used for text and time series data (1D signal) classification, 2D CNN for audio and image applications, 3D CNN for video, and volumetric data. With the advancements of low-cost computational power and 3D sensors, 3D computer vision is becoming increasingly commonplace in many industrial and user applications such as surveillance, industrial inspection, and health.

Typically CNNs require extensive datasets for training due to the large number of parameters that require optimisation. Techniques such as Transfer Learning [33] reduce the data volume demand. For example in 2D CNNs for image oriented tasks, models are pre-trained on the large ImageNet [34] dataset and subsequently re-trained on an application-specific dataset, the latter often smaller in size. A similar approach is reported by Hara *et al.* [35], [36] where a 3D CNN with random weight initialisation is trained on the Kinetics [37] dataset and transfer learning is explored on other datasets. The 3D convolution is obtained by convolving a 3D filter kernel through stacking multiple continuous frames to produce a 3D cube, allowing 3D CNNs to create hierarchical representations linking multiple consecutive frames to capture motion-related information [38]. However, the additional kernel dimension is at the expense of increased computational intensity.

The study reported in the paper compares the performance of two spatio-temporal CNN architectures to annotate subsea survey frame sequences, as illustrated in Figure 2, along with an evaluation of their size and efficiency in terms of training and evaluation of performance times. For the 2D model, each frame in the sequence is applied through a 2D CNN to create a prediction for every single frame. Given the confidence of the network varies from frame to frame, this approach can lead to sporadic False Positives (FP) and False Negatives (FN) which are physically not possible during the survey as the inter-frame spacing is short. Filtering such as averaging across the sequence can be applied post-prediction to mitigate against sporadic fluctuations and produce a final annotation. In the 3D case, the entire sequence is used as the input to a 3D CNN that outputs a single annotation, reducing fluctuations inherently.

Subsea video footage varies significantly across the pipeline length governed by diverse lighting conditions, bathymetry, sand and particle agitation, fouling, and vegetation, amongst others. These environments result in diverse contrasts and textures of the pipeline objects and events of interest. Instance Normalization (IN) has been widely used in Style Transfer [39]–[41] to make neural networks invariant to texture and style changes. Furthermore, Batch Normalization (BN) [42] is widely used to preserve content-related information. Instance Batch Normalisation (IBN) [8] combines these two techniques to increase the robustness of the neural network by filtering out sample-specific contrast information while preserving the content information. Results have shown a consistent improvement when the BN layers are replaced by IBN, motivating the development of 2D and 3D models that utilise a ResNet-50 [31] and 3D ResNet-50 [35] respectively, both modified to incorporate IBN layers.

The IBN ResNet architecture is expanded to use 3D convolution kernels that can capture both temporal and spatial information. Similar to the 2D IBN-ResNet50 [8], the layers of the model are changed to perform 3D convolutions, instance, batch normalization and pooling; a diagram of the 3D IBN-ResNet block is shown in Figure 3. An adaptive layer consisting of average and max pooling is introduced after the IBN-ResNet encoder, and then the features are flattened and concatenated before being inputted to two fully connected (linear) layers. Furthermore, BN and Dropout [43] layers are introduced between the linear layers to regularise the Head/Classifier.

IV. Dataset Description and Sampling

The raw data used are video streams from a North Sea survey conducted in 2012 covering 201 kilometres. The ROV camera provides a view of the crown of the pipe, looking slightly forward, as shown in Figure 1. The video data were provided in MPEG format with resolution of 576×704 and frame rate of 25 fps, with each video filenames containing the timestamp for the start of the inspection. Additionally, annotations created by trained Data Coordinators containing the event type and timestamps for the beginning and the end of each event were provided.

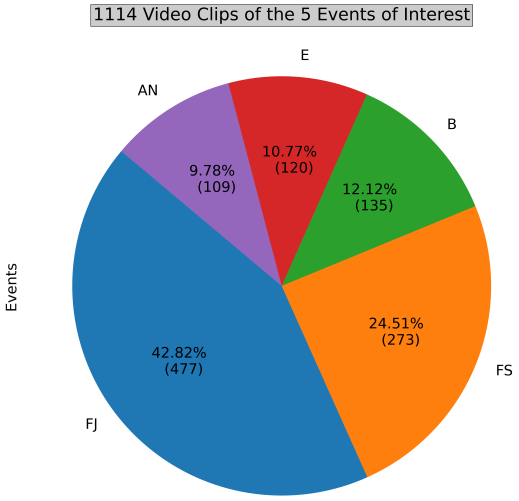

Fig. 4. Video Clip Distribution of the 5 Events of Interest

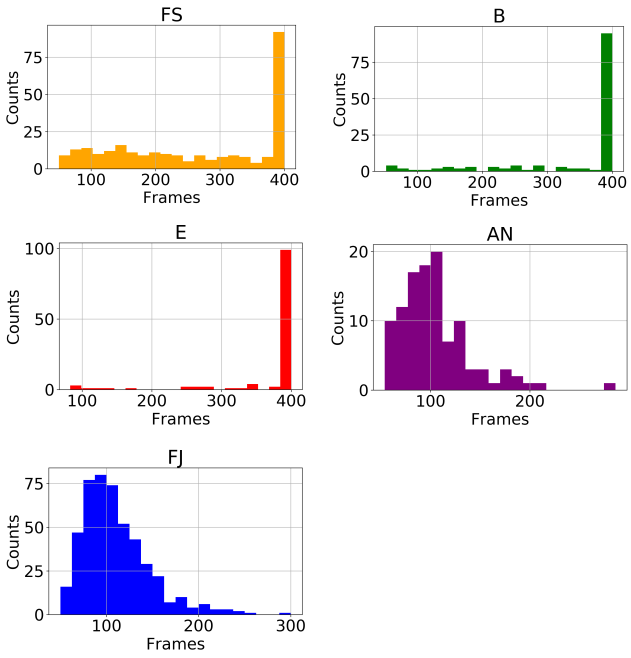

Fig. 5. Histograms of Frames per Event Video Clip

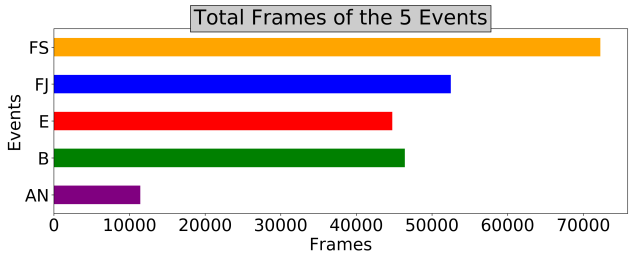

Fig. 6. Total Event Frames

A. Dataset Preparation

The event timestamps were used to extract frame sequences from the appropriate video file using OpenCV [44]. The events of interest and thus the labels are:

- *Burial (B)*: the pipeline is buried underneath the seabed and thus protected.
- *Exposure (E)*: the pipeline is exposed; visible and prone to damage. When the pipeline is exposed, other pipeline features/events become visible:
  - *Anode (AN)*: pipeline bracelet anodes are specifically designed to protect subsea pipelines from corrosion [45]. Data Coordinators visually recognise anodes by the banding that appears in the orthogonal direction of the pipeline; anodes have no surface vegetation growth.
  - *Field Joint (FJ)*: the point where two pipe sections meet and are welded together, typically occurring every 12 metres. Data Coordinators recognise Field Joints due to the depression on the pipeline surface.
  - *Free Span (FS)*: pipeline segments that are elevated and not supported by the seabed (either due to seabed erosion/scouring or due to uneven seabed during installation), pose significant risk to the asset; currents or moving objects (debris, nets and etc.) could damage the pipeline. FS are more apparent on the starboard and port video feeds; the centre camera is used to judge the seabed depth on the pipeline.

Examples of 4-frame sequences can be seen in Figure 1 (4-frame sequences are shown as an example, however the proposed 3D model utilises 16-frame sequences). The event distribution of the extracted video clips is shown in Figure 4. The data set contains 1114 video clips in total, consisting of 105 clips of burial and 979 clips of exposure, of which 120 clips are single *Exposure (E)*, 477 clips contain *Field Joints (FJ)*, 109 clips contain *Anodes (AN)*, and 273 clips contain *Free Spans (FS)*. A balanced sampling technique is implemented and described in subsection IV-B to tackle the high class imbalance of the dataset.

The histograms of frames per clip for the 5 events of interest are shown in Figure 5. The video clips of the 5 events have varying lengths with 50 and 400 frames being the minimum and maximum duration respectively. The short events of *FJ* and *AN* last an average of 3-4 seconds (75-100 frames), while the events of *B*, *E* and *FS* last up to 16 seconds (400 frames). The short events add some noise to the dataset because of the start and end timestamp annotations provided. For example, after experimentation it is observed that for an *AN* event lasting 100 frames, the anode is actually visible in the middle 70-80 frames. The extra 20-30 frames (i.e. one second) at the start and end of the event can be described as the transition period between these short events and the exposure of the pipeline before and after anodes and field joints are visible.

The combination of the event video clip distribution with the histograms of frames per event leads to Figure 6, which demonstrates the total frame distribution of the five events of

the dataset. The full dataset contains 227,334 frames. It is clear that the *AN* is a minority class whereas the *FS* is the majority class. The other 3 classes *(FJ, B, E)* have a similar amount of frames.

B. Balanced Sampling and Data Augmentation

Data resampling is commonly used in machine learning to balance datasets, by oversampling minority classes and undersampling majority ones [46]. By altering the relative frequencies of examples, dataset resampling enables the training of fairer models, which do not discriminate against minority classes. Oversampling adds repeated samples from minority classes, which could cause the model to overfit. To address this issue, oversampling is combined with image augmentation techniques. In this work, a balanced sampler has been used that ensures there is the same number of samples per event in a mini batch. For example, for a mini batch size of 10, the mini batch contains 2 samples from each event.

The 3D CNN is trained and tested on 16-frame sequences, whereas the 2D CNN is trained on single frames, but tested on 16-frame sequences by utilizing a rolling average of 16 predictions. To get meaningful inputs for both models, frames and sequences are sampled from the video clips. Not all frames of every clip are used because consecutive frames are highly correlated with each other and therefore add no extra information for learning.

Sampling frames or sequences differs for the events of long and short duration. For the events of long duration *(E, B, FS)* single frames and sequences are sampled every 8 frames. An example is illustrated in Figure 7 for a long event of 300 frames. For the short events *(AN, FJ)* exploratory data analysis has shown that the anode or field joint is usually visible in the middle of the video clip but sometimes not present in the initial or final frames. Therefore, a sampling method has been developed that simulates these events with a normal distribution with the mean μ being the middle of the event and a standard deviation σ of 15 frames. An example is provided in Figure 8 that indicates how the sampling is done in a short event of 120 frames. The light blue dots represent the starting frames of the sequences used to train the 3D IBN-ResNet50 model, but also the individual frames used to train the 2D IBN-ResNet50 model. Although the sampling is made in a similar manner, the final number of training data points is different for the two models because of the sequence length of the 3D case exceeding the length of some video clips. In the example of Figure 8, the frame-sequences close to the 80th frame are discarded.

To address the variability of ROV speed, instead of sampling directly 16-frame sequences, 50-frame (2 seconds) sequences are first sampled and then temporal augmentation is applied to convert to 16-frame sequences. Furthermore, consecutive frames are highly correlated because of the high frame rate and thus provide no extra information for learning. When augmentation is used, two temporal transforms are utilised.

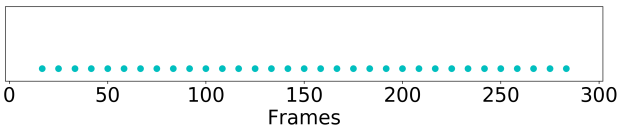

Fig. 7. Example of sampling frames and sequences from long Events of E, B and FS

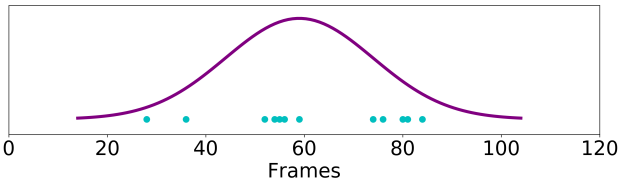

Fig. 8. Example of sampling frames and sequences from short Events of AN and FJ

- Inverse Order, the order of the sequence is sometimes inverted to address the issue of the ROV changing its direction of travel during a survey
- Temporal Elastic Transformation, which stretches or shrinks a video at the beginning, end or middle part to simulate the ROV speed change.

When no augmentation is used, 50-frame blocks are downsampled to 16-frame blocks which is common in video recognition [47] because of computational constraints. After getting either single frames or 16-frame sequences, spatial augmentations are applied to create a more diverse dataset and tackle susbsea image challenges such as camera movement, varying lighting conditions, contrast and blurriness from the seabed sand. During training, every sample is altered with one or two of the transforms listed below:

- Elastic Transformation which transforms the images by moving pixels locally around using displacement fields [48].
- Random Rotation by a maximum of 20 degrees angle to address the ROV camera rotation.
- Gaussian Blur to address images that contain sand from the seabed.
- Horizontal Flipping for tackling the overfitting on the layout words that the inspection software writes to the frames, and ensuring that a mirror of any event should also be detected as that.
- Increasing and decreasing the contrast and brightness to address the varying lighting conditions that subsea surveys contain.

The Vidaug [49] python library is used for the video augmentation. No spatial transforms are used during the validation and testing phases.

V. Training Configuration

Both single frames and frame sequences are labelled using a multi-label annotation approach since some of the events

recorded during the pipeline survey are not mutually exclusive. The pipelines are either buried underneath the seabed or exposed and thus visible. However, additional events such as field joints, anodes, and free spans can only be observed when the pipeline is exposed.

Multilabel image classification has been widely used in scene understanding [50]–[52], where multiple objects appear in an image, and thus more than one label can be assigned to a sample. In the multilabel setting, by applying a sigmoid after the last linear model, every element in the vector of the final 5 predictions is squashed in the range $(0, 1)$, with 0 being the negative class and 1 the positive; this is similar to performing 5 different binary classifications [53]. In this work, the two models are trained by optimizing the Focal Loss metric [54] as it offers better model calibration [55]. Focal Loss is defined as:

$$\text{Focal Loss} = -(1 - p_i)^{\gamma} \cdot \log p_i \tag{1}$$

where p_i is the predicted 5-score vector. The term $(1-p_i)^{\gamma}$, with the focusing parameter $\gamma \geq 0$, is a modulating factor to reduce the influence of correctly classified samples on the loss. In this work, γ is set to 2. With $\gamma = 0$, Focal Loss is equivalent to Binary Cross Entropy Loss. By using the focus parameter γ more weight is given to hard missclassified samples than to easy samples, so the contribution of each sample to the loss is different depending on the classification error.

Hard or noisy samples in the dataset exist because of wrong human annotations, camera and ROV movement as well as fish and vegetation. Another method used in this work to address the noise in the dataset is label smoothing. Label smoothing is a regularization technique for classification problems to prevent the model from predicting the labels too confidently during training and generalizing poorly. It is also used when there is a wrong label assignment in the dataset [56]–[58]. It is implemented by replacing one-hot encoded label vector y_{hot} with a mixture of y_{hot} and the uniform distribution:

$$y_{ls} = (1 - a) \cdot y_{hot} + a/K \tag{2}$$

where $K = 5$ is the number of labels, and a is a hyperparameter that determines the amount of smoothing. If $a = 0$, we obtain the original one-hot encoded y_{hot} . If $a = 1$, we get the uniform distribution. After experimentation, a was set to 0.1 in this work.

After the balanced sampling, the 2D CNN is trained with 16,149 samples, while the 3D is trained with 14,876. Although the sampling is performed in a similar manner for both architectures, the number of samples is different because some of the sequences sampled exceed the total length of the video clip and thus are discarded. The Adam optimizer [59], [60] is used for the training of both models with a cyclic learning rate policy [61], [62] with maximum learning rate of 0.001. The final models are saved based on the lowest validation loss. The 2D CNN is trained for 30 epochs and the model resulting in the lower validation loss is acquired on epoch 17, whereas the 3D CNN is trained for 50 epochs and the same model is acquired on epoch 27. The 2D CNN converges faster due to starting with ImageNet weights, while the weights of the 3D are randomly initialized. In addition, both models are trained with a similar amount of data, but the 2D model has half the size of its 3D counterpart. Two NVIDIA A40 GPUs are used for the training and the mini batch sizes are 40 and 10 for the 2D and 3D model, respectively. The validation and test sets are the same for both models and consist of 4,658 and 4,746 samples (frame sequences), respectively.

VI. Performance Evaluation

Before sampling individual frames or frame sequences, the video clips have been split into training (60%), validation (20%) and test (20%) sets in a stratified way based on the histograms of frames per event (Figure 5). This ensures that frames from a particular event only appear in one of these sets and thus there is no leak of information from one set to another. The sampling of single frames and sequences for the validation and test sets is done in the same way as described for the training set and illustrated in Figures 7 and 8. The difference is that neither resampling nor augmentation is applied in these sets other than the temporal downsampling of the frame sequences.

The validation set is used to ensure that there is no overfitting and to provide a test platform for searching an optimal confidence threshold for each label using Precision-Recall curves [63], [64]. For every 16-frame input, the 3D model produces one single output. In the case of 2D CNN, the threshold search happens after averaging the 16 confidence score predictions. The holdout test set, or unseen data, is used to assess the final performance of the model after the training and the selection of optimal thresholds. It provides an unbiased estimate of learning performance.

In this application, the evaluation metrics used for multi-label classification can be divided into two categories [65]; label-based metrics and example-based metrics. Label-based metrics evaluate each label separately. Therefore, any metric that can be used for binary classification can be used as a label-based metric. These metrics can be computed on individual class labels and these are reported in Tables I, II to indicate the challenges of predicting individual labels rather than relying on aggregated metrics that might not identify issues with particular classes. These are described by the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F1\text{-}Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{6}$$

The aggregated example-based evaluation metrics are designed to compute the average difference between the true

TABLE I
2D LABEL-BASED TEST SET METRICS

Event	Threshold	Accuracy	Precision	Recall	F1-Score	tn	fp	fn	tp
Exposure	0.518	0.923	0.947	0.957	0.952	722	201	160	3646
Burial	0.485	0.923	0.820	0.778	0.799	3649	157	204	719
Field Joint	0.655	0.969	0.925	0.913	**0.919**	3750	67	79	833
Anode	0.399	0.947	0.806	0.844	**0.824**	3904	139	107	579
Free Span	0.777	0.996	1.000	0.987	**0.993**	3403	0	17	1309

TABLE II
3D LABEL-BASED TEST SET METRICS

Event	Threshold	Accuracy	Precision	Recall	F1-Score	tn	fp	fn	tp
Exposure	0.657	0.935	0.943	0.978	**0.960**	698	225	80	3726
Burial	0.280	0.933	0.865	0.781	**0.821**	3694	112	202	721
Field Joint	0.424	0.893	0.714	0.748	0.731	3544	273	229	683
Anode	0.280	0.870	0.562	0.492	0.525	3780	263	348	338
Free Span	0.171	0.981	0.954	0.981	0.968	3341	62	24	1302

TABLE III
MODEL COMPARISON

Models	Parameters	EMR	Precision	Recall	F1-Score	Training Time (2 GPUs)	Inference Time (1 GPU)	Training Samples
2D IBN-ResNet50	**25.6** M	**0.853**	**0.908**	**0.905**	**0.901**	**5 mins per epoch**	185 ± 15 ms per 16 frames	16,149 single frames
3D IBN-ResNet50	45.5 M	0.725	0.879	0.878	0.865	95 mins per epoch	194 ± 15 ms per 16 frames	14,876 16-frame sequences

labels and the predicted labels. When average performance metrics are reported (Table III), the metrics are first calculated for each instance (example) and then averaged.

For aggregated accuracy, however, a stricter metric is used. Exact Match Ratio (EMR) extends the accuracy used in the single label setting for multi-label prediction. It does not distinguish between complete incorrect and partially correct results. All labels of a sample should be predicted correctly to count as a successful classification. Formally, the EMR is defined as

$$EMR = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y_i}) \tag{7}$$

where $I(y_i = \hat{y_i}$ is the indicator function equal to 1 only when every element in the round truth vector y_i is equal to every element in the prediction vector $\hat{y_i}$ and n is the number of input samples.

Precision is the proportion of labels predicted correctly to the total number of actual labels, averaged over all instances.

$$Precision = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i \cap \hat{y_i}|}{|y_i|} \tag{8}$$

Recall is the proportion of labels predicted correctly to the total number of predicted labels, averaged over all instances.

$$Recall = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i \cap \hat{y_i}|}{|\hat{y_i}|} \tag{9}$$

The definition of precision and recall naturally leads to the following definition for F1-measure (harmonic mean of precision and recall):

$$F1\text{-}Score = \frac{1}{n}\sum_{i=1}^{n} \frac{2|y_i \cap \hat{y_i}|}{|y_i| + |\hat{y_i}|} \tag{10}$$

VII. RESULTS

The validation set consists of 4669 samples and it is used for finding the optimal thresholds for the 5 labels by utilising Precision-Recall curves which balance the trade-off between False Positives and False Negatives. After setting these thresholds the models are evaluated on a test set which contains 4729 samples. The thresholds, as well as the label-based metrics acquired by the 2D and 3D IBN-ResNet50 models, are presented in Tables I and II, respectively. In these tables, the cells with the highest F1-score performance are in bold, because F1-score is chosen as the most important evaluation metric.

The results of Table I indicate that the 2D classifier for the events *(E, FJ, FS)* performs better. On the other hand, for the events *(AN, B)* lower F1-Score is achieved although the Accuracy is in higher levels. The reason that these events are more challenging than the others is they both belong to minority classes. For *Anodes*, this can be seen in Figure 6, while *Burial* is mutually exclusive with the *Exposure* label which in the balanced sampling accounts for the 80% the data.

A similar behaviour is recognised in the results of Table II for the *FS* event, while the performance of the *E* and *B* improves. For the events of *AN, FJ* the performance drops and the reason is that the transition from exposure to these classes can create confusion (noise) in the dataset and lead to either False Positives or False Negatives. Rolling average helps the 2D model to mitigate this issue, as one final prediction is made out of 16. A conclusion that can be extracted is that for events

with greater duration (*E, B*) the 3D model performs better than for short duration events such as *AN* and *FJ*. However, in general, the 2D model outperforms its 3D counterpart in the label-based evaluation metrics.

In the industrial subsea pipeline inspection, it is argued that it is preferable to overpredict an event than missing it. If necessary, this can be accommodated by decreasing the threshold for a particular event (e.g. AN). Precision-Recall curves offer a balance between False Positives and False Negatives; the threshold value is a parameter that can be manually changed depending on the sensitivity that is required for a specific application.

Finally, Table III provides an overall comparison of the two approaches presented in this work. The 2D and 3D CNN models are compared based on their example-based averaged EMR, Precision, Recall, and F1-Score, training, inference times, and samples used for training. The evaluation metrics indicate that the 2D model followed by a rolling average outperforms the 3D model. In addition, the training of the 2D CNN is significantly faster due to its fewer parameters (almost half than the 3D CNN) and the use of pretrained ImageNet-weights as initialization makes the convergence faster. However, the 3D model outperforming the 2D model in the events of *Exposure* and consequently *Burial*, in combination with the number of samples used for the training of both models, can lead to the conclusion that the 3D model is in need for more training samples because of its bigger size and more parameters.

The inference time is short and similar for both CNNs, while the memory usage each of them occupies in a single GPU for inference is less than 3 GB. Therefore, both models can be deployed on an ROV with an embedded GPU system.

VIII. Conclusions

This work provides an analysis and evaluation of two CNN-based spatio-temporal architectural paradigms towards automating subsea survey video annotation. Results indicate that the 2D model can outperform its 3D counterpart, , achieving an Exact Match Ratio of 85% and F1-Score of 90%, while being more efficient in training and having fewer parameters. In addition, label-based metrics indicate that the 2D model performs better in the case of short events of anode and field joint, while both models have similar performance in the long events of exposure, burial, and free span. A promising direction for further work is to investigate the tuning of frame sequence length while sampling and training to improve the performance of the 3D model. The impact of this work in practical contexts is that it develops the potential for an intelligent decision support tool to be used in conjunction with human operators to improve decision-making in the annotation process of subsea pipelines.

Acknowledgment

The work was partially supported by The Data Lab Innovation Centre, Edinburgh, Scotland, UK (project registration code 16270), the Oil and Gas Innovation Centre, Aberdeen, Scotland UK (project registration code 18PR-16) and N-Sea, Zierikzee, Netherlands. The Data Lab and the Oil and Gas Innovation Centres are funded by the Scottish Funding Council through the Innovation Centres Programme.

References

[1] M. Ho, S. El-Borgi, D. Patil, and G. Song, "Inspection and monitoring systems subsea pipelines: {A} review paper," *Structural Health Monitoring*, p. 147592171983771, apr 2019. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1475921719837718

[2] C. Mai, S. Pedersen, L. Hansen, K. L. Jepsen, and Z. Yang, "Subsea Infrastructure Inspection : A Review Study," pp. 71–76, 2016.

[3] Y. Guo, H. Li, and P. Zhuang, "Underwater Image Enhancement Using a Multiscale Dense Generative Adversarial Network," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 862–870, 2020.

[4] C. International, "2021 Virtual Research Exchange Asset inspection powered by computer vision : The use of deep neural networks for automating the detection and classification of pipeline external," no. March, 2021.

[5] F. R. Petraglia, R. Campos, J. G. R. C. Gomes, and M. R. Petraglia, "Pipeline tracking and event classification for an automatic inspection vision system," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. Baltimore, MD, USA: IEEE, May 2017, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/8050761/

[6] V. Bharti, D. Lane, and S. Wang, "Learning to Detect Subsea Pipelines with Deep Segmentation Network and Self-Supervision," *2020 Global Oceans 2020: Singapore - U.S. Gulf Coast*, 2020.

[7] A. Stamoulakatos, J. Cardona, C. McCaig, D. Murray, H. Filius, R. Atkinson, X. Bellekens, C. Michie, I. Andonovic, P. Lazaridis, A. Hamilton, M. Hossain, G. Caterina, and C. Tachtatzis, "Automatic annotation of subsea pipelines using deep learning," *Sensors (Switzerland)*, vol. 20, no. 3, 2020.

[8] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via ibn-net," 2020.

[9] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2017.

[10] M. Jacobi and D. Karimanzira, "Underwater pipeline and cable inspection using autonomous underwater vehicles," in *2013 MTS/IEEE OCEANS - Bergen*, June 2013, pp. 1–6.

[11] M. Jacobi and D. Karimanzira, "Multi sensor underwater pipeline tracking with AUVs," in *2014 Oceans - St. John's*. St. John's, NL: IEEE, September 2014, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7003013/

[12] V. Bharti, D. Lane, and S. Wang, "Robust Subsea Pipeline Tracking with Noisy Multibeam Echosounder," *AUV 2018 - 2018 IEEE/OES Autonomous Underwater Vehicle Workshop, Proceedings*, 2018.

[13] M. Narimani, S. Nazem, and M. Loueipour, "Robotics vision-based system for an underwater pipeline and cable tracker," in *OCEANS 2009-EUROPE*, May 2009, pp. 1–6.

[14] P. Zingaretti and S. M. Zanoli, "Robust real-time detection of an underwater pipeline," *Engineering Applications of Artificial Intelligence*, vol. 11, no. 2, pp. 257–268, April 1998. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0952197697000018

[15] G. Conte, S. Zanoli, A. M. Perdon, G. Tascini, and P. Zingaretti, "Automatic analysis of visual data in submarine pipeline inspection," in *OCEANS 96 MTS/IEEE Conference Proceedings. The Coastal Ocean - Prospects for the 21st Century*, vol. 3, Sep. 1996, pp. 1213–1219 vol.3.

[16] A. Ortiz, M. Simó, and G. Oliver, "A vision system for an underwater cable tracker," *Machine Vision and Applications*, vol. 13, no. 3, pp. 129–140, July 2002. [Online]. Available: http://link.springer.com/10.1007/s001380100065

[17] M. Asif and M. Rizal, "An Active Contour and Kalman Filter for Underwater Target Tracking and Navigation," in *Mobile Robots: towards New Applications*, A. Lazinica, Ed. I-Tech Education and Publishing, December 2006.

[18] A. Khan, S. S. A. Ali, A. Anwer, S. H. Adil, and F. Meriaudeau, "Subsea pipeline corrosion estimation by restoring and enhancing degraded underwater images," *IEEE Access*, vol. 6, pp. 40 585–40 601, 2018.

[19] M. Martin-Abadal, E. Guerrero-Font, F. Bonin-Font, and Y. Gonzalez-Cid, "Deep semantic segmentation in an auv for online posidonia oceanica meadows identification," *IEEE Access*, vol. 6, pp. 60 956–60 967, 2018.

[20] M. O'Byrne, V. Pakrashi, F. Schoefs, and a. B. Ghosh, "Semantic Segmentation of Underwater Imagery Using Deep Networks Trained on Synthetic Imagery," *Journal of Marine Science and Engineering*, vol. 6, no. 3, p. 93, Aug. 2018. [Online]. Available: http://www.mdpi.com/2077-1312/6/3/93

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[22] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, G. Kendrick, and R. Fisher, "Automatic annotation of coral reefs using deep learning," in *OCEANS 2016 MTS/IEEE Monterey*. Monterey, CA, USA: IEEE, Sep. 2016, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7761105/

[23] M. Jeon, Y. Lee, Y.-S. Shin, H. Jang, and A. Kim, "Underwater Object Detection and Pose Estimation using Deep Learning," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 78–81, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896319321718

[24] A. King, S. M. Bhandarkar, and B. M. Hopkinson, "A Comparison of Deep Learning Methods for Semantic Segmentation of Coral Reef Survey Images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Salt Lake City, UT: IEEE, Jun. 2018, pp. 1475–14 758. [Online]. Available: https://ieeexplore.ieee.org/document/8575347/

[25] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[26] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, "Robotic Detection of Marine Litter Using Deep Visual Detection Models," *arXiv:1804.01079 [cs]*, Sep. 2018, arXiv: 1804.01079. [Online]. Available: http://arxiv.org/abs/1804.01079

[27] W. Xu and S. Matzner, "Underwater Fish Detection Using Deep Learning for Water Power Applications," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, NV, USA: IEEE, Dec. 2018, pp. 313–318. [Online]. Available: https://ieeexplore.ieee.org/document/8947884/

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 779–788, 2016.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, December 2015, arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385

[32] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, 2020. [Online]. Available: https://doi.org/10.1007/s13748-019-00203-0

[33] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," 2018.

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

[35] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 6546–6555. [Online]. Available: https://ieeexplore.ieee.org/document/8578783/

[36] ——, "Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Venice: IEEE, Oct. 2017, pp. 3154–3160. [Online]. Available: http://ieeexplore.ieee.org/document/8265584/

[37] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," *arXiv:1412.0767 [cs]*, Oct. 2015, arXiv: 1412.0767. [Online]. Available: http://arxiv.org/abs/1412.0767

[39] Y. Gao, Y. Ding, F. Wang, and H. Liang, "Attentional colorization networks with adaptive group-instance normalization," *Information (Switzerland)*, vol. 11, no. 10, pp. 1–13, 2020.

[40] H. Nam and H. E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," *Advances in Neural Information Processing Systems*, vol. 2018-Decem, pp. 2558–2567, 2018.

[41] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 1510–1519, 2017.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," p. 30.

[44] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[45] H. Fang and M. Duan, "Submarine Pipelines and Pipeline Cable Engineering," in *Offshore Operation Facilities*. Elsevier, 2014, pp. e1–e181. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780123969774000068

[46] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, June 2002. [Online]. Available: https://jair.org/index.php/jair/article/view/10302

[47] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.

[48] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, vol. 2003-Janua, pp. 958–963, 2003.

[49] Okankop, "okankop/vidaug." [Online]. Available: https://github.com/okankop/vidaug

[50] A. Zeggada and F. Melgani, "Multilabel classification of UAV images with Convolutional Neural Networks," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. Beijing, China: IEEE, July 2016, pp. 5083–5086. [Online]. Available: http://ieeexplore.ieee.org/document/7730325/

[51] Y. Liu, L. Sheng, J. Shao, J. Yan, S. Xiang, and C. Pan, "Multi-Label Image Classification via Knowledge Distillation from Weakly-Supervised Detection," *2018 ACM Multimedia Conference on Multimedia Conference - MM '18*, pp. 700–708, 2018, arXiv: 1809.05884. [Online]. Available: http://arxiv.org/abs/1809.05884

[52] Y.-C. Chen, S.-F. Chen, C.-K. Yeh, and Y.-C. F. Wang, "Order-Free RNN with Visual Attention for Multi-Label Classification," p. 8.

[53] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *arXiv:1811.03378 [cs]*, November 2018, arXiv: 1811.03378. [Online]. Available: http://arxiv.org/abs/1811.03378

[54] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[55] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. H. Torr, and P. K. Dokania, "Calibrating deep neural networks using focal loss," 2020.

[56] L. Mao, "Label smoothing." [Online]. Available: https://leimao.github.io/blog/Label-Smoothing/

[57] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS, 2019.

[58] J. Hou, H. Zeng, L. Cai, J. Zhu, J. Chen, and K. K. Ma, "Multi-label learning with multi-label smoothing regularization for vehicle re-identification," *Neurocomputing*, vol. 345, no. February, pp. 15–22, 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.11.088

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[60] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ryQu7f-RZ

[61] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2017, pp. 464–472.

[62] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," *arXiv:1506.01186 [cs]*, June 2015, arXiv: 1506.01186. [Online]. Available: http://arxiv.org/abs/1506.01186

[63] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, pp. 1–21, 03 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0118432

[64] O. Gharroudi, H. Elghazel, and A. Aussem, "Ensemble Multi-label Classification: A Comparative Study on Threshold Selection and Voting Methods," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. Vietri sul Mare, Italy: IEEE, November 2015, pp. 377–384. [Online]. Available: http://ieeexplore.ieee.org/document/7372160/

[65] X. Z. Wu and Z. H. Zhou, "A unified view of multi-label performance measures," *34th International Conference on Machine Learning, ICML 2017*, vol. 8, pp. 5778–5791, 2017.