IADIS International Journal on Computer Science and Information Systems
Vol. 19, No. 1, pp. 15-30
ISSN: 1646-3692

DENSE SEMANTIC REFINEMENT USING ACTIVE SIMILARITY LEARNING

Connor Clarkson, Michael Edwards and Xianghua Xie
Computer Science Department, Swansea University, Swansea, United Kingdom

ABSTRACT

Defect detection has achieved state-of-the-art results in both localisation and classification of various types of defects, manufacturing domains is no exception to this. Just like in many areas of computer vision there is an assume of very high-quality datasets that have been verified by domain experts, however labelling such data has become an increasing problem as we require greater quantities of it. Within defect detection the variability and composite nature of defect characteristics makes this a time-consuming and interaction-heavy task with great amount of expert effort. We propose a new acquisition function based on the similarity of defect properties for refining labels over time by showing the expert only the most required to be labelled. We also explore different ways in which the expert labels defects and how we should feed these new refinements back into the model for utilising new knowledge in an effortful way. We achieve this with a graphical interface that provides additional information as data gets refined into a dense segmentation, allowing for decision-making with uncertain areas of the image.

KEYWORDS

Similarity Learning, Data Refinement, Active Learning, Defect Detection, Interactive, Acquisition Function

1. INTRODUCTION

Gathering large pools of data has become a relatively straightforward task, with many automated ways of obtaining various sources of data. The development of pattern mining and feature representation learning approaches which leverage large collections of observations has resulted in data becoming a prized resource in recent years. Labelling such data has become an exponential problem, being a time-consuming and interaction- heavy task that involves a great amount of user effort. This development has led to the rise of active learning as a semi-supervised alternative to data labelling, where a selection of samples is labelled to refine the models be- haviour. Many domains require skilled expert knowledge to label such data; including medical image analysis (Budd, Robinson & Kainz, 2021), manufacturing quality control (Sun et al., 2018), and even genomics research Zhang & Zhou (2006). In the context of the manufacturing industry, steel is a diverse and heavily used product with many different use cases. Variability in how steel is processed can result in various types of defects such as lamination, heavy scales and edge damage. Often these defects are highly variant in shape and characteristics, lighting issues in capturing the defect, and other types of artifacts not considered defects, like soot or water (Sun et al., 2018). These often result in a challenging environment within manufacturing, and therefore it is commonplace to have visual inspection systems to help ensure a level of quality in the final product. The setup and running of such systems are complex due to the many different ways we can manufacture such products, and as such the deployment of such inspection systems will often be adapted for their circumstances (Neogi, Mohanta, & Dutta, 2014). These systems will often have a classification component, which attempts to recognise the different kinds of defects present on the surface of the material so that suitable approaches can be taken later in the development pipeline to correct the issue, whether this is cutting out defective regions or repurposing the product. Such systems often rely on large datasets, with engineering domain experts providing ground truth labels for the training of supervised approaches, with labelling often falling into two camps; dense labelling of pixels within the images captured by the system, or sparse labelling using bounding boxes. The dense labelling provides a fine resolution label of the defect but requires significant input by the domain expert and so is often prohibitive. The bounding-box approach allows an expert to label the defect with less overhead but can introduce incorrectly labelled pixels to what should be a ground-truth target for supervised approaches to utilise (Xu, Bai & Ghanem, 2019). Defects that have forms of curvature, such as lamination or scratches, can be a challenge for a region of interest (ROI) based detection system, due to noise and artifacts around the defect. Other challenges for ROI-based systems include the composite nature of some defects that can formalize from micro defects, resulting in regions that have multiple defects. This leads to predictions with less than desired bounding boxes, and uncertainty in their location and classification. The nature of a bounding box also means that labelling is often not pixel-perfect, resulting in sampled observations which are incorrectly labelled.

In this work, we propose a data refinement strategy based on querying the similarity of embedding vectors with a human-in-the-loop approach to fix mistakes in bounding-box labelled datasets for steel defect detection. We uniformly sample patches of the labelled image and learn an embedding space of patch clusters. Querying the embedding space allows us to create a deep segmentation of the steel surface, which can then assist the inspection team.

2. RELATED WORK

2.1 Image Processing for Surface Defect Detection

Image processing plays an important role in surface defect detection in many manufacturing industries. Often contributing to aspects of quality control - which ensures product reliability - and safety. The task of defect detection is to automatically locate and classify defects on the surface of a material with a high level of precision and efficient complexity of the algorithm. These two goals of the task are often trade-offs of each other due to the approximation requiring more time and resources to compute (Redmon et al., 2016). In previous work, utilising image processing methods have been used to dynamically define a threshold which detects the outliers, Wang et al. developed a histogram of image patches to find differences between samples with classes of defects (Wang et al., 2018). A different threshold was learnt for each of the features via a random forest. A hard challenge of detect detection is the variability in scales and features, therefore effectively distinguishing the diverse nature can result in less than desired results. Work into the spatial domain by localising defects based on this variability have shown promising results. Choi et al had such work that used filter-based methods to explore defects on different scales. Good detection performance comes from this, however, the types of detects are restricted to one type known as a hole-like defects, limiting the robustness (Choi et al., 2017).

2.2 Deep Learning for Surface Defect Detection

Convolutional Neural Networks (CNNs) have become widely popular in many domains including defect detection, due to being able to learn robust local image features during training. One of the first CNN-based approaches used for quality inspection was used to detect cracks based on image patches of concrete, followed by a sliding window approach to follow the crack during deployment (Cha, Choi & Büyüköztürk, 2017). This work was added upon by Song et al, who proposed a method based on U-net which showed robustness to back- ground noise while detecting cracks (Song et al., 2019). To deal with multiple types of defects and get better localisation results work utilised from object detection have shown great performance (Liu et al., 2022; Ullah et al., 2018). In this new deep learning era two types of detection approaches have been proposed but focus on different ends of the trade-off between speed and accuracy. Two-stage detection uses a paradigm of high-level abstracts to fine-grained. This process attempts to improve recall with the high-level abstracts, then refines localisation in the fine-grained stage based on the high-level abstract learning. Within manufacturing a new structural visual inspection system uses this two-stage process with a method known as Faster Region-based CNN (faster R-CNN) (Cha et al., 2018). They showed good average precision on different types of surface defects, including those on steel. Test-time detection is another challenge in steel manufacturing due to the speed steel moves through production (Sun et al., 2018). As a result, test-time detection needs to be at least as fast as production speeds as to not introduce a bottleneck. (Li et al., 2018) proposed a real-time detection approach based on You Only Look Once, which can reach speeds of up to 83 FPS on cold-rolled steel surfaces. These detection approaches are known as one-stage detectors due to completing in one-step inference, however, performance is a challenge when dealing with objects that are dense and small such as within defect detection. In general, these different approaches have achieved very good performance but assume that the dataset has a large amount of high-quality labelled images, this is quite impracticable in an industrial manufacturing setting due to the impact of labour-intensive labelling practices on domain experts (Luo et al., 2020).

2.3 Similarity Learning

Similarity methods are a possible alternative to conventional supervised learning techniques, in which we train a model to learn what samples are similar and dissimilar based on a given metric which describes the similarity between two observations, often this metric is a form of distance. Distance is a useful way to measure similarity as the score is of continuous output form, allowing for a more fine-grained understanding of relationships between observations compared to discrete class labels. Since we focus on the relationships between our observations rather than explicit labels its more robust to errors in the labelling system. For these reasons similarity learning makes for a good use-case for training functional models or in cases where we need to transfer to a new task, due to the learnt similar features being richly placed close (Xiao et al., 2021).

The most prominent approach in recent literature is FaceNet (Schroff, Kalenichenko & Philbin, 2015), which uses a CNN to learn an embedding of pairs of faces. The work is based on the triplet loss, which optimizes the embedding space such that samples with the same label are closer to each other while those with different labels are pushed further away (Weinberger & Saul, 2009). To encourage faster convergence and better generalisation, FaceNet proposes an online variant of mining observed triplets based on a large batch size (Schroff, Kalenichenko & Philbin, 2015). The mining strategy involves finding valid triplets given a batch of embedding vectors based on selected anchors. We mine for useful positives and negatives from some metric, where the goal is to move the positives closer to the anchor while moving the negatives away. This creates clusters of similar features within the embedding space. (Hermans, Beyer, & Leibe, 2017) evaluated different variants of the triplet loss, finding that sampling the hardest triplets within a batch and applying a soft margin was the best for person re-identification. Using a suitable mining strategy is task dependent problem as we may have observations that are all very similar and we require a approach that finds nuance in the relationships of different observations. Using a mining strategy that selects triplets where the negative sample is close to the anchor and the positive sample is further away would be more beneficial for learning as we want to find the observations that are the currently considered the most dissimilar but are actually similar, thus once corrected leads to bigger learning jumps between triplets.

Beyond person re-identification learning similarities naturally becomes useful in image retrieval domain such as in place recognition were given an image of a location, we want to retrieve images that are locally close to where this image was taken (Arandjelovic et al., 2016; Ribeiro et al., 2018). A prominent approach to this is NetVLAD, which proposes a trainable VLAD layer that can be used in deep visual place recognition pipelines for fine-tuning to a task Arandjelovic et al. (2016), Arandjelovic & Zisserman (2013). Utilising the triplet loss in a pre-training scheme benefits from placing similar features close together in a more explicit way, allowing quick return on similar but also dissimilar samples. In general similarity learning gives the benefit of finding pairwise relations between unlabelled objects from the feature similarities and providing a level of robustness due to the transitive property between the anchor and positive pairs. The main challenge of similarity learning is the mining of triplets on large datasets, due to the growth in complexity when mining triplets as the dataset grows in size.

2.4 Active Learning

Active refinement and analysis are widely explored domains that leverage user input to handle low-confidence predictions and then feed the changed annotations made by the user back into the model (Branson et al., 2014; Vezhnevets, Buhmann & Ferrari, 2012). The main principle of active learning involves finding the samples that will gain the newest knowledge for the model, known as the acquisition function. Most approaches involve finding k samples to relabel based on an uncertainty metric, followed by a subsequent training session. This cycle of labelling and then training completes the framework of active learning. By leveraging user input, active learning assists in model development by providing insight into hard samples the model struggles with, allowing for better focus on complex regions of the domain. This process can also be a two-way interaction between user and model, with users benefiting from seeing which samples the model is having problems with, which can be a way of interpreting the current version of the model during the refinement cycle. The main driving force in active learning is through the acquisition function, identifying samples that warrant further user insight, and measuring which samples the user should label next is commonly categorized into three buckets; uncertainty (Blundell et al., 2015; Gal, Islam & Ghahramani, 2017), sample representation (Sener & Savarese, 2017), and training effects (Settles, Craven & Ray, 2008). Once a session of active refinement is complete, an update of model weights incorporates the new knowledge. This is commonly done by updating labels of the whole dataset but can also be achieved by populating a growing database of samples that have been observed (and potentially refined) by the user. Both of these ideas are explored in this work.

3. METHOD

Our framework consists of a new acquisition function based on embedded vectors, where we use mined triplets of anchors, positives, and negatives to refine a pre-labelled dataset of images containing surface defects on sheets of steel. During the labelling phase, the user updates the labels of the top five hardest positive and negative samples via a graphical interface. We explore three different types of initial mask labels, one using domain-expert labelled bounding boxes (ROI), a dense segmentation provided by an off-the-shelf pre-trained U-Net architecture, and the other being a uniformly random mask. User refinements are then incorporated into the learning strategy by either updating the original label set, or by developing a new set of samples which have been observed by the user during the active learning loop.

The following sections describe the triplet loss, the mining strategies, and finally how we refine the labels within our dataset.

3.1 Triplet Loss

Images are embedded in a d-dimensional Euclidean space, which is represented by $f(x) \in R^d$ where x is an image. A triplet consists of an anchor x_i^a, a positive which has the same label as an anchor x_i^p and a negative which has a different label to the anchor x_i^n. The goal of this loss is to ensure that the distance between the anchor and the negative is greater than the distance between the anchor and the positive over all possible triplets. Therefore, the loss L minimised is as follows,

$$L = \sum_i^N d\left(f(x_i^a), f\left(x_i^p\right)\right) - d\left(f(x_i^a), f(x_i^n)\right) + \alpha$$

Here, d is a metric function, in our case, this is the Euclidean distance between the embedding vectors of the anchor and either the positive or negative. N is the number of samples in a batch and α is a margin used to enforce a distance between positive and negative pairs. Generating all possible triplets would result in some triplets that already satisfy our goal and therefore do not add much to learning, it is becoming exponentially more expensive as the size of the dataset increases. Therefore, we deploy a strategy to mine different and useful triplets. The following section discusses a few of these strategies.

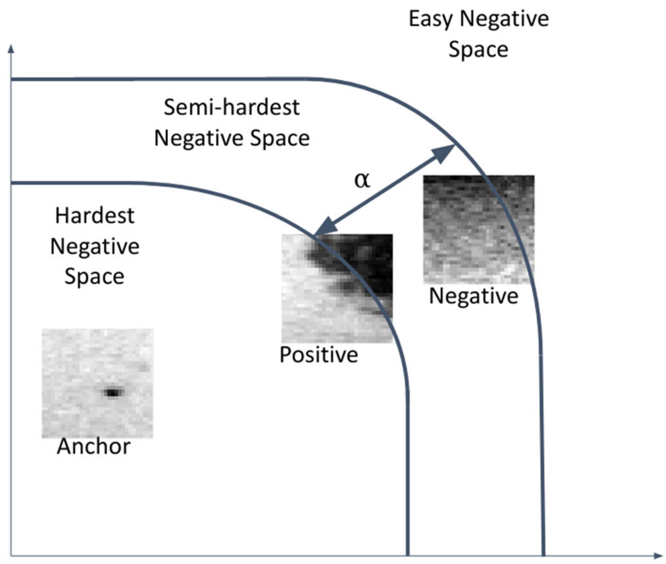


Figure 1. Negative Samples that are closer to the anchor than a positive are within the hardest negative space. Semi-hard negatives are between the positive and a α margin. Negative samples from this space are easier than hardest negatives as they are not as close to the anchor so the triplet loss will not be as great. We avoid easy negatives as they provide no new knowledge to the model and therefore return a loss of 0

3.2 Active Online Mining Strategies

In online mining strategies, we compute meaningful triplets for each batch during the training process. The benefits of this are threefold. Firstly, mining the dataset in a batch typically leads to better generalisation and smoother learning (Schroff, Kalenichenko & Philbin, 2015), Secondly, if the dataset has some mislabeled data, this would dominate the mining process. This is because if a sample was labelled negative incorrectly then the model is correct in putting it close to its anchor, yet the mining strategy would consider this a hard negative and thus the loss would try to push that sample away from the anchor. We utilise this benefit in our acquisition function. Finally, due to the change in embedding space as the model learns, the triplets would change between being hardest to semi-hardest to easy triplets (Hermans, Beyer & Leibe, 2017). If x_i^p is closer to x_i^a plus some α margin than the distance between x_i^a and x_i^n, then this is considered an easy triplet as it already meets the following condition,

$$\left|\left|\boldsymbol{f}(\boldsymbol{x}_i^a) - \boldsymbol{f}(\boldsymbol{x}_i^p)\right|\right|_2^2 + \boldsymbol{\alpha} < \left|\left|\boldsymbol{f}(\boldsymbol{x}_i^a) - \boldsymbol{f}(\boldsymbol{x}_i^n)\right|\right|_2^2$$

Easy triplets do not add much new knowledge to the model as they already meet the criteria and therefore should be avoided. Instead, we focus on selecting triplets that break the condition in equation 2. Given that we select positives such that $\operatorname{argmax}_{x_i^p} \left\|f(x_i^a) - f(x_i^p)\right\|_2^2$, we select negatives that meet the following condition,

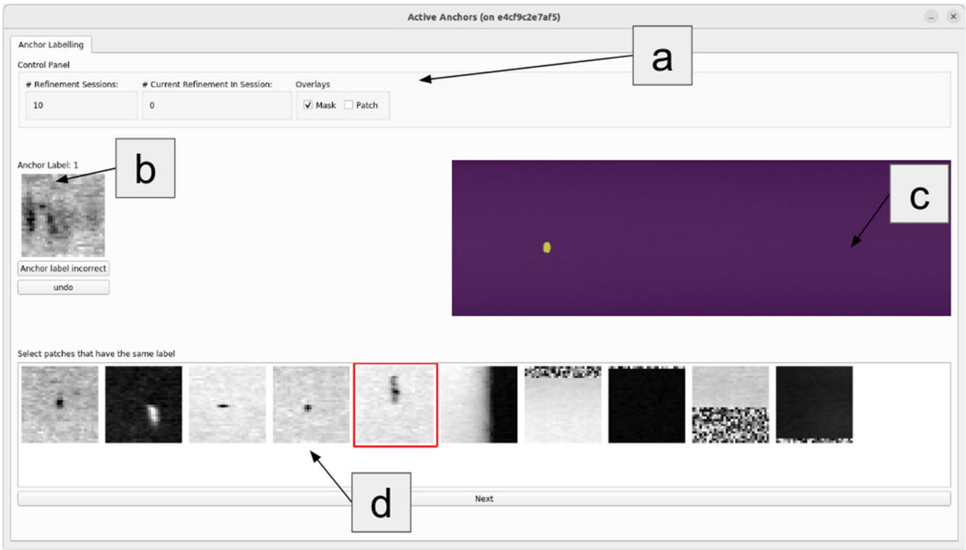
$$\|f(x_i^a) - f(x_i^n)\|_2^2 < \left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2$$

These are known as the hardest triplets, as we select the closest negative to the anchor. Always selecting the hardest possible images for the model to learn from is a complex task and leads to a difficult learning environment, due to always having the largest possible loss per batch. To ease this task, we can select negatives such that,

$$\begin{gathered}\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 < \\ \|f(x_i^a) - f(x_i^n)\|_2^2 < \\ \left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 + \alpha\end{gathered}$$

We consider these triplets semi-hard, as the negative is between a positive and the margin. They are not hard negatives as the positive is closer to the anchor, yet they are also not easy negatives as they are not beyond the margin. Throughout this mining process, for a given anchor we will always select the hardest positive such that,

$$\operatorname{argmax}_{x_i^p} \left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2$$



Active Anchors (on e4cf9c2e7af5)

Control Panel

# Refinement Sessions: 10

# Current Refinement In Session: 0

Overlays: ☑ Mask □ Patch

Anchor Label: 1

Select patches that have the same label

Figure 1. a: The control panel consists of the number of refinement sessions done, the number of current refinements done within a session and a mask overlay and an option to show where the selected patch is located within the image. Users can select either the anchor or any of the 10 patches in d and the user will see where that selected patch is located in c. b: The selected anchor of this batch. The user can update the anchor's label and undo the actions performed. c: Display the image of the currently selected patch. d: Users select patches that they think have the same label as the anchor

Each of the strategies depends on how we select the negative relative to the anchor and the positive. Figure 1 demonstrates the different mining strategies.

As we need to evaluate the distance between the anchor and every other patch within the batch, we can then also order them to find the k-worst patches. These patches would be considered those the model is struggling with the most, and therefore we query these patches to the user with the graphical interface shown in Figure 2.

3.3 Dataset Label Refinement

Within our framework, we utilise an online patch-generation procedure that is based on three different types of initial masks; expert-labelled ROI, a pre-trained segmentation and a uniform random mask (Figure 3). The expert-labelled ROI masks consist of one to three boxes that interact the area where a defect could be. These masks have a large merge of error as more than one type of defect can exist within the bounding box, some defects are not labelled, and finally many bounding boxes do not cover the whole defect. The segmentation mask allows for a pre-training scheme where given the expert-labelled ROIs find common the features - such as similar types of defects and group them together for labelling. This grouping creates a segmentation, however uncommon defects are often labelled as background class. Uniform random mask acts as our worst-case in which the most refinement is needed, this can also be considered as observation with no labels. Each of these masks act as minor dataset differences seen in manufacturing. During the labelling phase, the user is shown an anchor patch selected at random or based on model entropy. We then use a mining strategy, to find hard positives and semi-hard negatives creating our triplets. The user interacts with the model via the graphical interface shown in Figure 2. Users are shown a selected anchor patch and its corresponding 5 hardest positive and hardest negative patches per the similarity scoring in equations 2 and 3. The user selects the candidate patches that they think have the same label as the anchor, effectively either agreeing with the model's prediction or correcting its labelling. After a batch of refinements is carried out, and the underlying labels updated, the model goes into the subsequent training phase. This cycle of labelling and then training defines our framework.

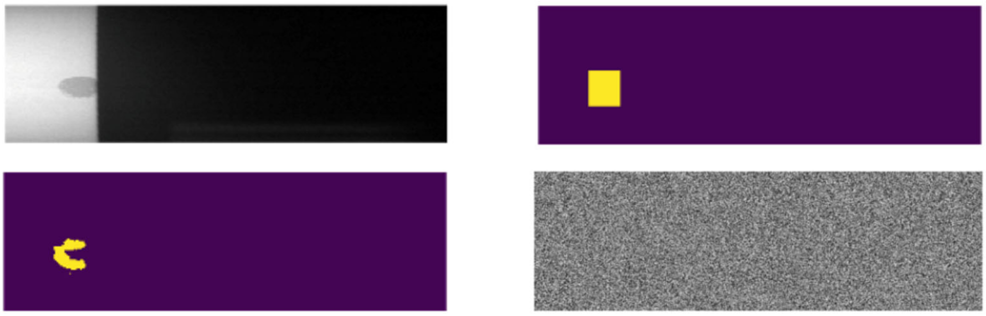


Figure 2. From right to left: Example raw steel that contains at least one defect. An ROI label of the defects are represented as a mask. A pre-trained segmentation mask of input. A uniform random mask which acts as our worst-case for refining. Masks allow us to uniformly sample pixel coordinates via the label, which can then be used to extract patches

3.4 Expert Interactions with GUI

In designing an active learning experiment, we require to define how experts should interact with a system and how a system should use this new knowledge. The former is described in this section. User interactions are preformed via the graphical interface shown in Figure 2. The first type of interaction is a grouping task. An anchor patch is selected randomly within the batch of patches or by the patch that the model is most uncertain about, via entropy. The worst patches compared to the anchor are then shown. The user selects the patches they believe are similar to the anchor, correcting the model. In the second interaction users still require to group patches, however for the ones they believe are similar a shifting operation is performed. Users shift the centre pixel of the patch with the aim of putting the defect in the middle of the patch. If we train the model on user verified patches then this reduces spatial variability in the dataset, allowing for a cleaner learning curve.

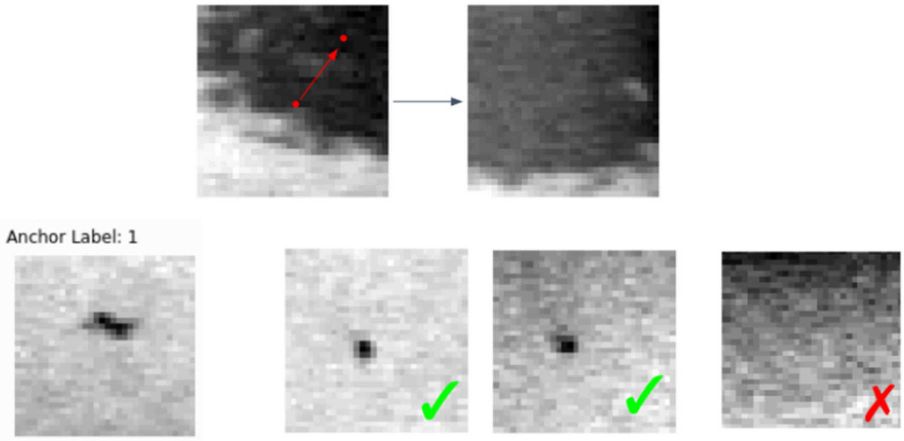


Figure 3. Experts have two ways to interact with our framework to refine the data. The bottom approach is a grouping task where the user compares the anchor patch to other patches that the current iteration of the model is having trouble with. Users select patches they believe to be the same as the anchor. In the top approach a user still needs to group similar patches together with the anchor patch but then performs the extra step of shifting. This moves the centre position of the patch by middle clicking a pixel

4. APPLICATION

As an evaluation case study, we apply our refinement approach to the domain of surface inspection within steel manufacturing, due to the labelling challenges this domain has; namely non-defect artifacts, bounding-box- based labelling difficulties, and the domain expertise required. The following sections discuss the dataset used to evaluate our framework, followed by how we implemented our approach and deployed our experiments.

4.1 Dataset

We evaluate our framework on grayscale images of steel, captured during the cold rolling manufacturing pro- cess. The dataset consists of 5000 images, with each containing between one and three defects. There are multiple different types of defects within this dataset, but in this instance, we categorise this as a binary classification problem, identifying defect/non-defect on a per-pixel basis. Images are captured at various angles and positions along the mill, creating variance in both scene illumination and placement of the steel sheet within the camera's field of view. For the initial labelling, we utilise bounding box labelled regions of interest, a U- Net architecture or a uniform random label. The bounding box labels are created by domain experts from a live system. Due to the nature of the defect geometry, ROIs can often provide sub-optimal labelling, with positives labelled as negative, and vice-versa. By contrast, the pre-trained segmentation model provides dense labelling which can handle the varying shapes of the defect but is reliant on suitable generalisation and performance of the utilised model for the specific application domain.

4.2 Implementation

For our implementation, we train a modified version of ResNet with 15 layers (He et al., 2016) returning a predicted binary label and embedding vector of the input given. We uniformly sample patches based on the selected mask type and then use hard mining to find positives and semi-hard mining for negative patches to form our triplets. In refinement learning is it more important to focus on the training than the validation loss as the training set is what is being refined over time. We use a reasonably large batch of 128 triplets, allowing for more available data to mine, and approach to assist in finding the hardest possible triplets which leads to an increased inter-class embedding distance (Dong & Shen, 2018).

Once training has converged, we move to the labelling phase with the graphical interface. The user is asked to select patches they think look similar to the anchor. These patches are considered the worst due to their distance away from the anchor. Patches that the user selected that have a different label to the anchor are updated via the mask. Patches that have the same label but are not selected get updated as well. Users continue this process until 50 refinements have been made, which then triggers the training phase. As part of the experiments, we explore three different ways of showing the refined labels to the model. The first is via our patch generation procedure, where we update the mask and then re-sample that mask during training. The second is that we show only patches that have been refined by the user via the selection of similar patches to the anchor, similar to online incremental learning. Our third approach is based on correcting the patches of defects by selecting a defect pixel and we shift the patch to the new centre, which will then get added as a refinement.

We deploy our experiments to explore how the model should be shown the refined labels with the three different initial masks, these are via mask updates and only refined patches. Each set of experiments also looks into anchor selection as these patches are what define a mined positive and negative sample, therefore we explore anchors that the model finds uncertain as well as randomly selected. The following section discusses what we found during our experiments.

5. RESULTS

5.1 Quantitative Analysis

Quantitatively evaluating refinements in a semi-supervised setting is challenging as we have no way of confirming if the refinement is correct and how good it is without expert domain knowledge. Our only way to measure these results is based on the initial sparse masks within our test set due to the refinement only being performed and saved on the train set. To generate our test set we use 1113 images of steel and select 128 patches uniformly per image, we then compare the mask label from the patch coordinate with the prediction from the final version of the model. This results in allowing us to compute the F1, Recall and Precision for each experiment.

In each experiment the user is shown patches to refine based on the selected anchor. In table 1 the user selects patches similar to the anchor and the update is applied to the central pixel, but in table 2 the user corrects the labelling by selecting the centre of the defect if one exists in the presented patch. In table 3 we perform the same labelling interaction as in table 1 but instead of the centre pixel being refined, we refine each pixel within the patch. Each contains two ways of feeding the model new information by updating the underlying mask that samples are drawn from, or only using refined samples to build a dataset of user-verified samples. Our results show that updating the mask (table 2) provides the highest quantitative metric scores against the pre-refinement labelling; however, the pre-refinement labels can be either over- or under-approximations of the actual ground truth, reinforcing the need for qualitative analysis to inspect the impact of refinement on the labelling of the underlying data. Generally, user refinement provides low recall while having a higher precision, we theorise that this is because the model is not shown the global context of the dataset which mask update provides. As we uniformly sample defects from all over the image during training. With this global context from the mask update, we see that recall is normally higher than precision. Generally, our approaches work better with the initial segmentation from the U-Net, this is presumably due to the start being more faithful to the geometry of the defect in comparison to the bounding box approach. Using uniform random labels as our initial mask is the most challenging of our experiments as we do not provide any knowledge to the framework as the labels are meaningless and thus can be considered as if we are refining the label from the ground up. Refinement in this experiment to a suitable performance is possible but it takes a very long time to complete as many of the initial refinements progress is very small due to many background patches labelled as defects. As a result, some iterations of the model can over-fit to background either only training of varied user patches or on mask only. Updating the mask via a whole patch refinement as shown in table 3 generally performs well but not as good as refining a single pixel of the patch. We also found that labelling whole patches converges to this worse performance compared to single pixel refinement a lot quicker. This is because labelling a single pixel is less information to the model than a whole patch and secondly, a single pixel refinement allows for more nuance knowledge to the model that is often more valuable to the learning process. Providing this nuance knowledge shows the model that there is more to learn and that is impacted in the loss metrics.

Table 1. Mask update refers to refinements made to the mask over time. During training, we uniformly sample coordinates from the mask that relate to the centre pixel of our patches. User Refined refers to the labels of patches that the user changed in the graphical interface, while observed are labels of patches that the user agrees with

	Initial Mask								
	ROI			Pre-trained Segmentation			Uniform Noise		
	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
Mask Update	0.91	0.96	0.86	**0.96**	0.95	0.97	0.14	0.18	0.12
User Refined	0.96	0.93	**0.99**	0.72	**0.96**	0.58	0.64	0.80	**0.54**
Mask Update with Entropy Selection	**0.97**	**0.97**	0.98	0.95	0.93	**0.98**	0.18	0.22	0.15
User Refined with Entropy Selection	0.90	0.82	**0.99**	0.27	0.21	0.39	**0.67**	**0.98**	0.51
Mask Update & User Refined	0.79	0.88	0.72	0.69	0.74	0.65	0.35	0.63	0.24
Mask Update & User Refined With Entropy Selection	0.83	0.89	0.77	0.74	0.78	0.70	0.41	0.64	0.30

Table 2. Labels in these experiments get updated by the user shifting the centre of the patch to the defect via a graphical interface, where the user is shown the 5 hardest negatives and positives based on the anchor. We then compare patch prediction with a initial mask to compute quantitative results

	Initial Mask								
	ROI			Pre-trained Segmentation			Uniform Noise		
	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
Mask Update	**0.97**	0.95	0.98	0.68	0.90	0.55	0.10	0.12	0.09
User Refined	0.83	0.72	**0.99**	0.89	0.81	**0.99**	0.68	0.78	0.61
Mask Update with Entropy Selection	0.91	**0.96**	0.87	0.91	**0.92**	0.90	0.29	0.34	0.25
User Refined with Entropy Selection	0.86	0.77	**0.99**	0.89	0.82	0.97	**0.72**	**0.81**	**0.64**
Mask Update & User Refined	0.96	**0.96**	0.97	**0.94**	0.90	**0.99**	0.46	0.69	0.34
Mask Update & User Refined With Entropy Selection	0.96	0.95	0.97	0.83	0.81	0.85	0.49	0.72	0.37

Table 3. Labels in these experiments get updated via every pixel within a patch. During training these newly updated masks are uniformly sampled to get coordinates that are then used to extract patches

	Initial Mask								
	ROI			Pre-trained Segmentation			Uniform Noise		
	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision
Mask Update	0.70	**0.98**	0.54	**0.96**	**0.96**	**0.97**	0.25	0.32	0.20
Mask Update with Entropy Selection	0.74	**0.98**	0.59	0.80	0.89	0.72	0.24	0.31	0.19
Mask Update & User Refined	0.79	0.84	**0.75**	0.57	0.62	0.52	**0.32**	**0.47**	0.24
Mask Update & User Refined With Entropy Selection	**0.80**	0.89	0.72	0.72	0.71	0.73	**0.32**	0.36	**0.29**

5.2 Qualitative Analysis

To meaningfully show the quality of the refinement process we need some qualitative analysis as shown in Figures 5 and 6. We use a single image of steel along with its initial ROI mask, each row then contains a refinement session, until we get to the last row which is the final version of the model for that experiment. We see that the model does refine better for experiments where we just feed refined examples. Often if we uniformly sample from the mask, we get a circular segmentation around and filling the ROI, which is why we often see higher precision in these experiments. Selecting the anchor based on the highest entropy often leads to worst results, we believe this is due to the task for the model to learn being harder. This is because we mine the hardest negatives and positives from the hardest anchor in a batch. Our approach is also able to find defects that have not been labelled while also ignoring non-defect artifacts like luminance. We can also see that area within the ROI also gets refined such as in figure 5 (right) where we have two defects labelled as one.

In figure 6 we show that by adding more user interaction into the training our refinement will get better, this has the caveat that more time from domain experts is needed. Each of these patches are shown with an overlay displaying the pixel confidence of a defect, with ROI based approaches we see that there is a level of uncertainly around the box due to many of the pixels labelled as positive are actually negative. We find that the best way to deal with this uncertainly is to perform pixel labelling and shifting patches so that the defect is in the centre, this results in better ROIs and dense segmentations.

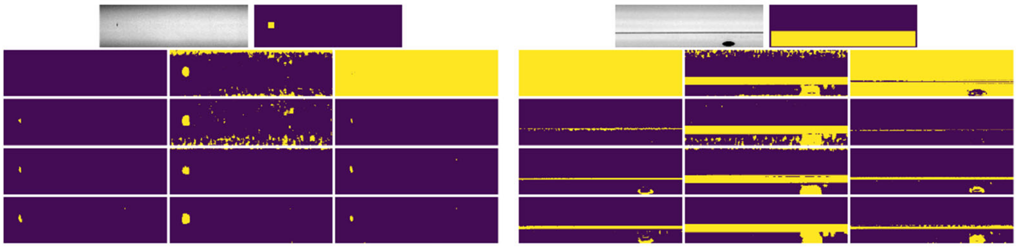


Figure 4. We show a forward pass of a single image of steel producing a dense segmentation over refinement sessions with three different experiments. This is accomplished via the graphical interface where the user is given the 5 worst negatives and positives based on an anchor, then selects new centre pixel for that patch. Row one displays the image and the initial mask. Column one is based on user refinement, column two is mask updates, and column three is user refined with entropy anchor selection. Each row shows a subsequent refinement after a session.

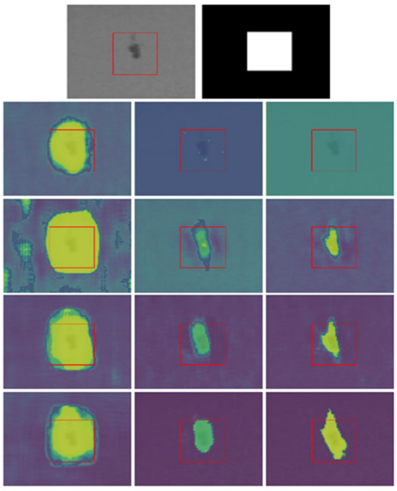


Figure 5. Zoomed in crop of the dense segmentation over refinement sessions when users select defect centers from within presented samples. Red box indicates initial ROI labelling, heatmap shows confidence of model output for positive defect detection. From top to bottom: steel image and initial ROI mask, refinement passes 1-4. From left to right: update of label mask only, user verified dataset, and user verified data with entropy-based anchor selection

6. CONCLUSION

In this paper, we demonstrate a new active learning framework for querying patches of images based on the triplet loss. Using initial masks, we mine for the worst negatives and positives in relation to a selected anchor. Masks are used to uniformly sample centre patch coordinates creating our batch. We also explore three different ways of feeding the model refinements during training. Our first method investigates refining the mask itself to improve the ROI into a dense segmentation. Secondly, only patches that have been refined and observed by the user, create an extra challenge for the model as our initial dataset is small but grows incrementally and the patches are considered the hardest. While the mask update is considered an easier approach as it gives a global context of the image allowing for the sampling of patches not considered as hard. The final method aims to alleviate the uncertainty around ROI by centring the patch to the object.

We utilise our framework within the defect detection domain with a focus on steel during the manufacturing process due to the highly variant nature of defects, adding to the complexity of ROI approaches that lead to uncertainty around defects. Refining such datasets require highly skilled expert knowledge. Our framework helps speed this process up by only querying the most needed patches to be refined.

ACKNOWLEDGEMENT

Funded by the EPSRC Centre for Doctoral Training in Enhancing Human Interactions and Collaborations with Data and Intelligence Driven Systems (EP/S021892/1).

REFERENCES

Arandjelovic, R. et al., 2016. NetVLAD: CNN architecture for weakly supervised place recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Las Vegas, USA, pp. 5297-5307.

Arandjelovic, R. and Zisserman, A., 2013. All about VLAD. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*. Portland, USA, pp. 1578-1585.

Blundell, C. et al., 2015. Weight uncertainty in neural network. *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France, pp. 1613-1622.

Branson, S. et al., 2014. The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization. *International Journal of Computer Vision*, Vol. 108, pp. 3-29.

Budd, S., Robinson, E. C. and Kainz, B., 2021. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, Vol. 71, 102062.

Cha, Y.-J., Choi, W. and Büyüköztürk, O., 2017. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 32, No. 5, 361-378.

Cha, Y.-J. et al., 2018. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 33, No. 9, 731-747.

Choi, D.-c. et al., 2017. Detection of pinholes in steel slabs using gabor filter combination and morphological features. *ISIJ International*, Vol. 57, No. 6, pp. 1045-1053.

Dong, X. and Shen, J., 2018. Triplet loss in Siamese network for object tracking. *European Conference on Computer Vision (ECCV 2018), Vol. 11217*. Munich, Germany, pp. 459-474.

Gal, Y., Islam, R. and Ghahramani, Z., 2017. Deep Bayesian active learning with image data. *Proceedings of the 34th International Conference on Machine Learning*. Sydney, Australia, pp. 1183-1192.

He, K. et al., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Las Vegas, USA, pp. 1578-1585.

Hermans, A., Beyer, L. and Leibe, B., 2017. *In defense of the triplet loss for person re-identification*. https://doi.org/10.48550/arXiv.1703.07737

Li, J. et al., 2018. Real-time detection of steel strip surface defects based on improved yolo detection network. *IFAC – PapersOnLine*, Vol. 51, No. 21, pp. 76-81.

Liu, Y. et al., 2022. Surface defect detection of steel products based on improved yolov5. *IEEE 41st Chinese Control Conference (CCC)*. Hefei, China, pp. 5794-5799.

Luo, Q. et al., 2020. Automated visual defect detection for flat steel surface: A survey. *IEEE Transactions on Instrumentation and Measurement*, Vol. 69, No. 3, pp. 626-644.

Neogi, N., Mohanta, D. K. and Dutta, P. K., 2014. Review of vision-based steel surface inspection systems. *EURASIP Journal on Image and Video Processing*. 2014, No. 50, pp. 1-19.

Redmon, J. et al., 2016. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Las Vegas, USA, pp. 779-788.

Ribeiro, P. O. C. S. et al., 2018. Underwater place recognition in unknown environments with triplet based acoustic image retrieval. *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Orlando, USA, pp. 524-529.

Schroff, F., Kalenichenko, D. and Philbin, J., 2015. FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. Boston, USA, pp. 815-823.

Sener, O. and Savarese, S., 2017. Active learning for convolutional neural networks: A core-set approach. *ICLR 2018 Paper*. https://doi.org/10.48550/arXiv.1708.00489

Settles, B., Craven, M. and Ray, S., 2008. Multiple-instance active learning. *Advances in Neural Information Processing Systems*, Vol. 20, pp. 1289-1296.

Song, L. et al., 2019. Weak micro-scratch detection based on deep convolutional neural network. *IEEE Access*, Vol. 7, pp. 27547-27554.

Sun, X. et al., 2018. Research progress of visual inspection technology of steel products - a review. *Applied Sciences*, Vol. 8, No. 11, 2195.

Ullah, A. et al., 2018. Pedestrian detection in infrared images using fast RCNN. *Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. Xi'an, China, pp. 1-6.

Vezhnevets, A., Buhmann, J. M. and Ferrari, V., 2012. Active learning for semantic segmentation with expected change. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*. Providence, USA, pp. 3162-3169.

Wang, Y. et al., 2018. Distributed defect recognition on steel surfaces using an improved random forest algorithm with optimal multi-feature-set fusion. *Multimedia Tools and Applications*, Vol. 77, No. 13, pp. 16741-16770. https://doi.org/10.1007/s11042-017-5238-0

Weinberger, K. Q. and Saul, L. K., 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, Vol. 10, No. 2, pp. 207-244.

Xiao, T. et al., 2021. Region similarity representation learning. *IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, Canada, pp. 10519-10528.

Xu, M., Bai, Y. and Ghanem, B., 2019. Missing labels in object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Zhang, M.-L. and Zhou, Z.-H., 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 10, pp. 1338-1351.