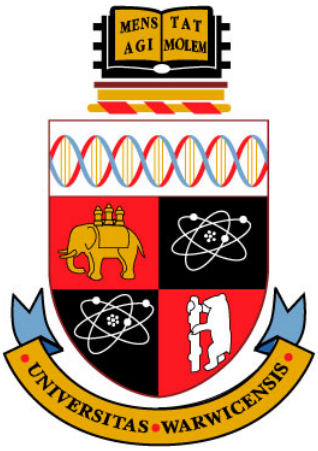

CS344 Final Dissertation

On the Robustness of Segmentation Models Used in Computational Pathology

Sam Molyneux

Supervisor: Dr. Fayyaz Minhas

Department of Computer Science

Contents

Acknowledgements

I would like to express my profound gratitude to my supervisor, Fayyaz Minhas, whose invaluable guidance has been outstanding. Fayyaz has consistently gone above and beyond his responsibilities as a supervisor, meeting with me weekly to discuss progress and offer indispensable advice. The project was his idea, and without him, it would not have been possible.

I am also profoundly grateful to Rebecca Dilipen for her ongoing encouragement and support throughout the project. She played a significant role in helping me refine and practice my presentation, ensuring it was the best it could be. She generously allowed me to use her laptop for the presentation, which was greatly appreciated.

I would like to thank Alex Foote, the creator of REET. The toolbox was necessary for my experiments.

I would also like to extend my gratitude to Dang Vu, who very kindly showed me that what I was trying to do was impossible! His expertise saved me a tremendous amount of time.

My appreciation goes to Mariusz Ceglarek for providing valuable feedback on my presentation, which greatly contributed to the writing of this dissertation.

I would like to extend my heartfelt thanks to my colleagues who were also supervised by Fayyaz: Luana Georgescu, Yoel Kastro Morlevi, Stephanie Christodoulou, and Hugo Boland. Throughout the project, we have supported one another, offering advice, feedback, and motivation. This mutual encouragement has enriched my academic experience and fostered an environment of collaboration and perseverance.

Abstract

Nuclear segmentation of histopathology images is an essential step in the digital pathology workflow. The advent of deep learning has led to the development of convolutional neural networks, which can automate segmentation. These networks can perform tasks more rapidly and, in some cases, more accurately than trained pathologists. As a result, nuclear segmentation models are being integrated into the clinical workflows of the NHS.

However, clinical environments can vary significantly between facilities, posing a challenge for these models. There is limited research on the robustness of segmentation models against clinical variations, which can be attributed to the lack of diverse testing data.

To address these limitations, we have developed image transforms that aim to simulate the most challenging conditions that nuclear segmentation models are likely to face. We then conducted a study evaluating the robustness of HoVer-Net and U-Net to these transforms.

Our experiments demonstrated that HoVer-Net and U-Net exhibit good robustness to the transforms, but they lack robustness to adversarial attacks. Furthermore, we found that adversarial training was not an effective method for improving robustness to the pathology-based transforms, suggesting that adversarial training does not improve the generalisability of nuclear segmentation models.

Contributions

- We showed that U-Net exhibits good robustness to variations in digital pathology.
- We showed that HoVer-Net exhibits good robustness to variations in digital pathology.
- We showed that adversarial training does not improve the robustness of segmentation models to variations in digital pathology.

Index

This section defines the key terminology used throughout this paper. Some definitions have been taken from [1] or [2] and then adapted for our purposes.

Pathology: "The branch of medicine concerned with the cause, origin, and nature of disease, including the changes occurring as a result of disease" [3]. In this dissertation, we will mainly use pathology to refer to the study of disease through tissue analysis, known as histopathology.

Digital Pathology: The systems used to digitise pathology slides and their associated storage, review and analysis.

Whole Slide Image A digital image of a histopathological glass slide captured at a microscopic resolution.

Computational Pathology(CPath): The process of utilising AI to extract information from digitised pathology images.

Adversarial Attack The process of manipulating an image that will be fed to a machine learning model, with the intention of making the model perform its task incorrectly on the manipulated image.

Perturbation A small alteration to an image.

Transform Any function applied to an image, e.g. perturbations, adversarial attacks, rotations, changing colour.

Robustness An ML model's ability to produce accurate and consistent results when its input images have been adversarially attacked, perturbed or transformed.

Feature Data from an image that a model uses to make a prediction. A feature is 'robust' if it is not altered by perturbations.

Chapter 1

Background

Histopathologists examine biopsies and pieces of tissue to aid in the diagnosis of diseases [4]. If there is a suspicion that a patient has cancer, a biopsy can be taken from the suspected tissue and analysed under a microscope. A Histopathologist can then reach a diagnosis by identifying changes in cells that are characteristic of disease [5].

Alternatively, a Whole Slide Image (WSI) of the tissue can be generated by a digital scanner and inputted into a neural network. The neural network can then classify the image as cancerous or not. This process of using Artificial Intelligence (AI) to extract information from digitised pathology images is called Computational Pathology (CPath) [1].

The expected global rise in cancer incidence [6] exacerbates the existing shortage of experienced pathologists in the UK [7] and other countries [8][9][10]. CPath seeks to alleviate these challenges by utilising advanced machine learning algorithms to increase the efficiency of pathological analysis, minimise diagnostic errors, and improve patient outcomes [11].

Convolutional neural networks (CNNs) are the most commonly used AI image analysis tools in CPath [12]. CNNs are used for supervised learning tasks, in which they are trained on labelled input data to learn a mapping function from input images to output labels [13]. The CNNs that we will use in this project are divided into three categories based on their mapping functions:

- **Classification:** A CNN takes an input image and, from a set of finite output labels, assigns a label to the image. For example, Figure 1.1 shows a classification CNN that takes a WSI image patch as input and outputs *cancerous* if it detects cancer and *not cancerous* if it does not.
- **Semantic Segmentation:** A CNN takes an input image and, from a finite set of output labels, assigns a single label to each pixel. We call this output a type map. For example, Figure 1.2 shows a semantic segmentation CNN that takes a WSI image patch as input and assigns one label to each pixel.
- **Nuclear Instance Segmentation:** A CNN takes a WSI image patch as input and outputs a nucleus instance map. The process begins with the algorithm assigning a unique label to each detected nucleus in the image. Next, the nucleus

instance map is created by assigning each nucleus pixel to one of the unique labels, indicating which nucleus the pixel belongs to. Nucleus instance segmentation models frequently provide a type map as well. Figure 1.3 shows the workflow of a typical nucleus instance segmentation model.

Recently, CNNs have been shown to outperform trained pathologists in some tasks [14][15]. Amid recent advancements in CPath, there has been a push for the integration of CNNs and other CPath solutions into clinical laboratories in the UK [16].

In June 2018, the NCRI Cellular Molecular Pathology initiative [17] and British In Vitro Diagnostics Association [18] organised a workshop that brought together academic, clinical, regulatory, and industry leaders from the field of pathology and digital pathology [16]. The workshop aimed to identify the criteria AI tools must fulfil to obtain regulatory approval for clinical deployment. The consensus opinions were collated from the workshop and outlined in a roadmap [16]. The roadmap states that AI will transform clinics in the next decade. The earliest uses will be the integration of machine learning image analysis tools into the routine clinical histopathology workflow.

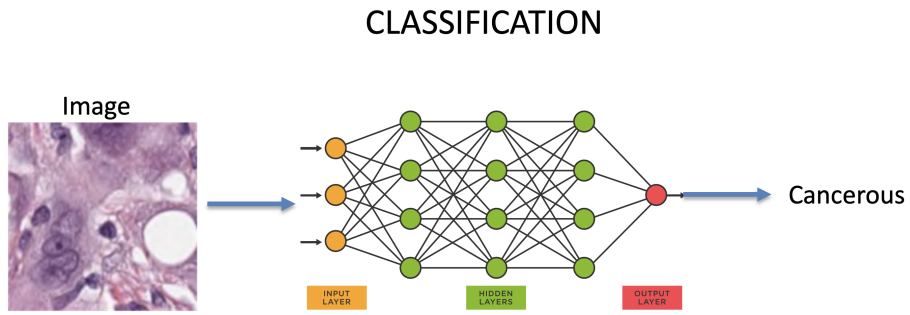

Figure 1.1: This figure shows the workflow of a classification CNN. The image on the left shows a patch of a WSI. The image patch is passed into the CNN and classified as cancerous.

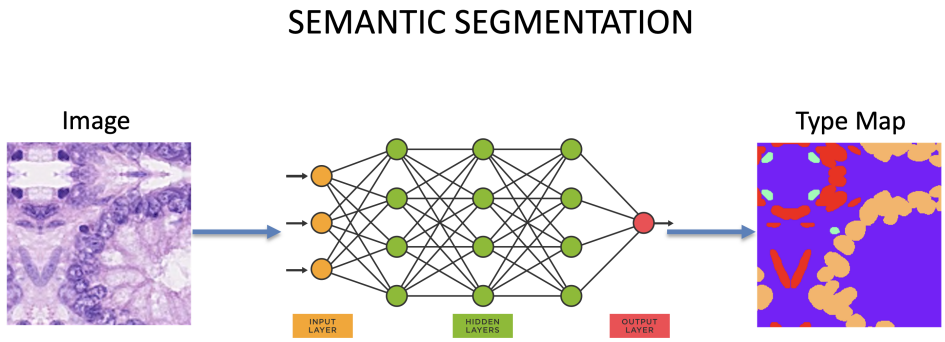

Figure 1.2: This figure shows the workflow of a semantic segmentation CNN. The image on the left shows a patch of a WSI. The image is passed into the CNN and each pixel is classified, with each class represented by a colour.

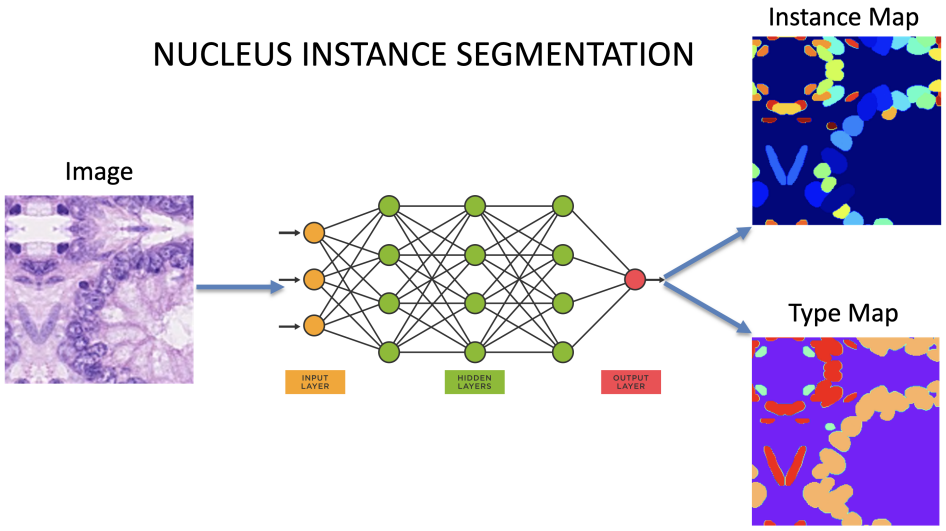

Figure 1.3: This figure shows the workflow of a typical nucleus instance segmentation CNN. A WSI image patch is inputted into the CNN, and the CNN produces a type map and an instance map. Each nucleus in the instance map is given a unique colour to represent its label; some colours look similar, but they are distinct.

In September 2018, Innovate UK awarded £50 million to create five new centres of excellence for digital pathology and imaging using AI [19]. The centres are known as the PathLAKE consortium. The centres aim to speed up diagnosis and improve patient outcomes by implementing CPath into clinical practice [16].

In August 2020, an additional £50 million was awarded to PathLake centres and NHS trusts to boost diagnostic capabilities for digital pathology using AI [20]. According to the press release, the funding will be used to deliver digital upgrades to pathology and imaging services across 38 NHS trusts and benefit 26.5 million people.

In March 2023, Ibex [21], Paige [22] and Aiforia [23] secured PathLAKE contracts to deploy AI-based CPath tools to NHS trusts in a clinical setting [24][25][26].

Chapter 2

Problem Statement

We have shown that AI will be used in NHS pathology labs to aid in the diagnosis of diseases. However, potential legal implications exist for a pathologist signing out a report using AI [16]. Before integrating an algorithm into the main report, pathologists must have confidence in the algorithm's output. The roadmap outlined in [16] stresses that before implementing CPath in diagnostic workflows, researchers must provide empirical evidence demonstrating the robustness of AI applications.

Robustness refers to a model's ability to produce consistent and accurate results in a wide variety of environments. Robustness is a requirement for CPath models, as they must perform effectively across many clinical settings. Variations between clinics can stem from: differences in tissue preparation, the digital scanners used for image generation, tissue thickness, and image acquisition methods [27]. Additionally, slides are stained using Hematoxylin and Eosin (H&E) to enhance the visibility of pertinent structures [28]. However, the lack of a standardised staining protocol leads to considerable variability in staining results across clinics, which has been identified as a significant source of variation [29].

The standard approach for improving a machine learning model's robustness is to train the model on diverse data. However, varied labelled histopathology data is not available. Labelling WSIs is slow and expensive, so datasets are often small and sometimes restricted to tissues taken from a handful of individuals from a single location.

To address this, Foote et al. [30] developed REET for classification models. REET applies adversarial augmentations to images and then evaluates model performance on the augmented images. The augmentations simulate the diverse settings that exist between clinics. For example, REET can simulate different staining procedures on an image and evaluate model performance on the stained images.

For a given image, REET searches for augmentations that result in the model misclassifying the image. This is why we refer to these augmentations as 'adversarial'. By using adversarial augmentations, we can evaluate model performance in the most challenging conditions the model is likely to face.

Foote et al. used this toolbox to show that CPath classification models lack robustness to many adversarial augmentations in REET. The models lack robustness to transforms such as: changing staining colour, zooming, changing brightness, rotation and more.

We have highlighted the importance of evaluating the robustness of CPath models, and we have shown that REET can be used to evaluate CPath classification models. However, semantic segmentation and nucleus instance segmentation models are also commonly used in CPath [31]. Before deploying semantic and nucleus instance segmentation models into the clinical setting, we must evaluate their robustness. That is why, in this paper, we have used the augmentations in REET to evaluate the robustness of U-Net [32] for semantic segmentation and HoVer-Net [33] for nucleus instance segmentation.

Foote et al. also used REET to show that some CPath classification models lack robustness to adversarial attacks [34]. Adversarial attacks are small image perturbations that are imperceptible to humans, but can change a model's classification of an image [35]. Figure 2.1 shows an example of a Projected Gradient Descent (PGD) [35] adversarial attack on a ResNet-18 [36] CNN, a model commonly used for image classification.

To demonstrate the extent to which classification models lack robustness to adversarial attacks, we will describe an experiment we performed on a ResNet-18 model. We trained the model to classify images as cancerous or not cancerous. We evaluated the model on a testing set of 712 images selected randomly from fold-1 of the PanNuke dataset [37]. This testing set consisted of 436 cancerous images and 276 non-cancerous images. We then performed a PGD attack on each image in the testing set as described in [35], with intensity changes limited to a maximum of 7 per pixel. Before the attack, the model correctly classified 88% of the images. After the attack, the model did not classify a single image correctly, despite the perturbed and original images appearing identical to humans. The model's vulnerability to the attack raises concerns about its robustness. We must now question which image features the model uses for classification.

If a model classifies visually indistinguishable images differently, it must not use the same features as a pathologist to detect cancerous cells. If the model uses different features, how do we know that those features can reliably predict cancer in diverse

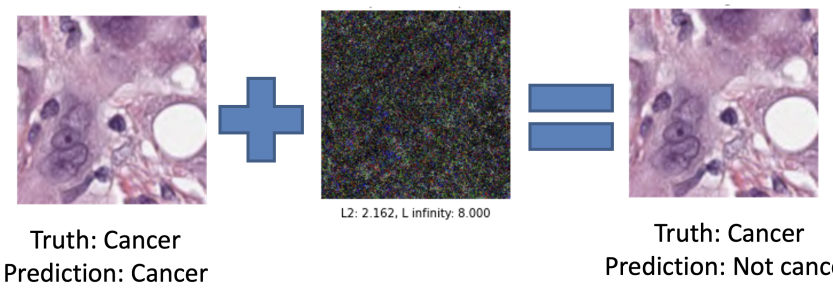

Figure 2.1: This figure shows an example of a PGD attack on a ResNet-18 model. The leftmost image shows the patch before the adversarial attack, and the ResNet-18 correctly classifies it as cancerous. The middle image shows the absolute value of the perturbation produced by the attack; we scaled the perturbation to make it visible. The rightmost image shows the patch after the adversarial attack. The perturbation has been added to the original image to produce the rightmost image. The ResNet-18 incorrectly classifies the rightmost image as not cancerous.

environments? We hypothesise that the augmentations in REET break the invisible features that the model uses to classify images. This is why CPath models lack robustness to adversarial augmentations.

If a model is robust to adversarial attacks, then we suspect that the model cannot be reliant on invisible features. Therefore, we hypothesise that improving robustness to adversarial attacks will also improve the robustness of the model to the pathology-based augmentations in REET. Since REET augmentations are designed to simulate variations in pathology, a more robust model would imply a more generalisable model. That is why, in this paper, we will also perform experiments to determine if increasing robustness to adversarial attacks can improve robustness to the augmentations in REET.

Chapter 3

Existing Work

3.1 Adversarial Attacks

We developed algorithms to optimise the augmentations in REET for segmentation models. The algorithms were adapted from existing algorithms for adversarial attacks. This section will briefly describe the algorithms used for adversarial attacks.

Szegedy et al. [38] were the first to show that imperceptible image perturbations could fool Deep Neural Networks (DNN). Consider a model M, image x, perturbation ρ, target class k and classification function f_M. The framework proposed by Szegedy et al. [38] states that the minimal perturbation ρ, required to make model M classify image $x+\rho$ as class k, is found by minimising $\|\rho\|_2$ subject to the following conditions:

1. $f_M(x+\rho) = k$.
2. The image $x+\rho$ remains within the valid pixel range.

This framework underpins all adversarial attacks. However, this is an NP-hard problem [2], so attacks focus on developing and solving suitable approximations.

Szegedy et al. proposed a constrained L-BFGS algorithm [39]. These attacks consistently made DNNs misclassify images, but were too computationally expensive to produce on a large scale.

Goodfellow et al. [40] proposed the Fast Gradient Sign Method (FGSM) to produce perturbations more efficiently. FGSM produces a perturbation ρ using the following formula:

1. $\rho = \epsilon \cdot \text{sign}(\nabla_x J_\theta(x, k))$

Where $J_\theta(x, k)$ represents the cost function for a model with parameters θ and a target class k, given the input image x. The gradient with respect to x is given by ∇_x, ϵ is the learning rate and sign() denotes the sign function applied individually to each element in the feature vector. The gradient $\nabla_x J_\theta(x, k)$ is calculated efficiently via backpropagation, and the algorithm takes a single step, allowing for rapid production. Because FGSM is fast to compute, it became more feasible to evaluate the robustness of models to adversarial attacks. FGSM allowed for the development of adversarial training, as we will discuss later in the chapter.

The Basic Iterative Method (BIM) [41] extends the idea of FGSM by allowing the algorithm to take multiple steps. Madry et al. [35] remarked that the BIM is a specific case of Projected Gradient Descent (PGD). PGD is the name we will use to describe all basic iterative algorithms for the rest of this dissertation, including the Iterative Least Likely Class Method (ILCM) [35], which is an untargeted version of BIM.

The general algorithm for PGD is as follows. After i iterations, PGD produces a perturbed image $x_{adv}^{(i)}$ with the following algorithm:

1. $x_{adv}^{(0)} = x + \text{Uniform}(\delta, p)$
2. $x_{adv}^{(i+1)} = \text{argmin}_z \|z - (x_{adv}^{(i)} + \epsilon \cdot \text{sign}(\nabla_x J_\theta(x_{adv}^{(i)}, k)))\|_p \quad \text{subject to} \quad \|z - x\|_p \leq \delta$

Where δ is a limit we place on the size of our perturbation and $\text{Uniform}(\delta, p)$ is a randomly selected perturbation ρ such that $\|\rho\|_p \leq \delta$. PGD uses backpropagation to update the perturbation iteratively. Thus, we can think of PGD as training the weights of a perturbation, much like we would train the weights of a neural network. PGD is the most commonly used attack in the literature due to its exceptional effectiveness and computational efficiency [2].

Other attacks serve different purposes. The Jacobian Saliency Map Attack (JSMA) [42] limits the perturbation by restricting the number of modifiable features, aiming to change as few pixels as possible. Su et al. [43] took this idea to the extreme with one-pixel attacks. They showed that, with a surprising degree of success, DNNs can be fooled by altering a single pixel. These attacks can be particularly helpful in evaluating whether a model places a large emphasis on a small number of features.

Carlini and Wagner's attacks [44] operate similarly to box-constrained L-BFGS attacks, but were shown to defeat defences that worked for box-constrained L-BFGS attacks. They advanced this work by developing an algorithm that can provably find the smallest perturbation required to fool a model [45]. This algorithm can be exceptional at evaluating exactly how robust a model is, but is impractical due to its computational cost.

Universal perturbations [46] not only fool a model on a specific image, but can fool the model when applied to any image. Universal perturbations are useful for evaluating robustness because they provide insight into which imperceptible features a model is using for a specific classification.

We wrote this section with the help of [2] and [47].

3.2 Augmentations and Other Transforms

Augmentations are widely used in training neural networks to simulate additional data and improve generalisability [48]. All of the following augmentations are commonly used in machine learning: rotation, flipping, cropping, zooming, colour transformations, adding brightness and adding random noise.

AutoAugment [49] attempts to improve random augmentations by using reinforcement learning to select the best augmentations for training the model. Adversarial AutoAugment [50] improves AutoAugment by producing augmentations that significantly improve generalisation, but with much lower computation times than other

methods. These augmentations aim to produce images that the model will perform poorly on, and are valuable tools for evaluating the robustness of models in challenging settings.

In CPath, stain normalisation is frequently used to prevent the model from using staining as a classification feature [51]. Stain normalisation involves applying staining colour adjustments to all images in the dataset, resulting in a consistent staining colour across all samples. Furthermore, staining augmentations can be used to detect if the model is using staining as a classification feature. Tellez et al. [52] developed a staining augmentation called HED, which simulates the variations in H&E staining. In a subsequent study [29], the authors compared different staining augmentation methods and discovered that HED was the best method for improving model generalisability to staining variations.

Adversarial stain mixing is a staining augmentation that was introduced in REET [53]. The advantage of using this method is that it is differentiable, so PGD can optimise the staining mixing augmentation.

REET introduced tools to search for pathology-based adversarial augmentations. In this way, we can find pathology-based augmentations that target the vulnerabilities of a model more directly. In [34], the authors used these tools to demonstrate that some CPath classification models lack robustness to variations that commonly occur in digital pathology.

3.3 Improving Robustness to Adversarial Attacks

Adversarial training is widely considered to be the best method for improving a model's robustness to adversarial attacks [2]. This method consists of performing an adversarial attack on each image before training. Adversarial training forces the model to learn robust features. The main disadvantage is that adversarial training is much slower than regular training.

3.4 Adversarial Robustness to Improve Generalisability

Stutz et al. [54] showed that robustness to adversarial attacks and generalisability are not contradicting goals. This work disputed earlier hypotheses, which claimed that increasing a model's robustness to adversarial attacks would reduce its accuracy [55][56]. Further work has even suggested that model generalisability is required for adversarial robustness [57].

To the best of our knowledge, there is no existing work exploring the relationship between the generalisability of nuclear segmentation models and their robustness to adversarial attacks.

3.5 Toolboxes for Evaluating and Improving Robustness

The Adversarial Robustness Toolbox [58] is an extensive toolbox of adversarial attacks and defences for classification models. However, the toolbox cannot evaluate the robustness of nuclear segmentation models and lacks pathology-based augmentations.

Foolbox [59] and AdverTorch [60] are very similar to The Adversarial Robustness Toolbox and suffer from the same limitations. The Robustness Toolbox [61] contains adversarial attacks and augmentations to evaluate the robustness of CNNs, but does not contain pathology-based augmentations.

3.6 Robustness of CPath Models

There is limited research on the robustness of CPath models. Foote et al. [34] used REET to show that some CPath classification models lacked robustness to PGD attacks. They then showed that the same models lacked robustness to other transforms in REET [53], such as stain mixing and HED.

Laleh et al. [62] showed that Vision Transformers (ViTs) could perform similarly to CNNs in CPath classification, but with much higher levels of robustness to PGD and FGSM attacks.

Schömig-Markiefka et al. [63] developed 12 augmentations to simulate artefacts that commonly occur in histopathology. They found that all artefacts negatively impacted their model's ability to detect prostate cancer. However, their work did not include optimisation methods for the artefacts.

Our work differs from previous work by focusing on nuclear segmentation models and optimising the augmentations to simulate challenging environments.

3.7 REET

REET is a robustness evaluation and enhancement toolbox for classification models in digital pathology. The toolbox can evaluate the robustness of CPath classification models against two transform types: differentiable and non-differentiable. Here we use 'transforms' to describe adversarial attacks and adversarial augmentations. For the rest of the paper, we will use 'transforms' for all REET augmentations. Figure 3.1 shows examples of all the transforms in REET.

3.7.1 Differentiable Transforms

These transforms have weights that can be updated via backpropagation. The differentiable transforms in REET are as follows:

- **Pixel**: This transform is produced identically to the BIM. The weights form a matrix with an identical shape to the image. The transform is applied by adding the matrix to the image.
- **Stain Mixing**: This is an adversarial stain mixing augmentation developed by Foote et al. [30]. The weights form a 3×3 staining matrix. Stain Mixing aims to simulate variations in staining.
- **Brightness**: This transform adds brightness to the image. The only weight is a single scalar. The scalar is added to the intensity value of every pixel in the image. Brightness aims to simulate variations in brightness and tissue thickness.

3.7.2 Non-Differentiable Transforms

Non-differentiable transforms have parameters that cannot be updated via backpropagation. The non-differentiable transforms in REET are:

- **Rotate**: This transform rotates the image. The only parameter is an angle. The image is rotated anticlockwise by the angle. The angle is measured in degrees. Rotate aims to simulate the tissue slide at different orientations.
- **Crop**: This transform crops a rectangle from the image. The cropped image is then padded with zeros to return it to its original shape. The parameters are: the coordinates of the rectangle's top left corner (x, y), the height of the rectangle and the width of the rectangle.
- **Blur**: This transform applies a Gaussian filter to a rectangular section of the image. The parameters are: the coordinates of the rectangle's top left corner (x, y), the height of the rectangle, the width of the rectangle, the Gaussian kernel size and the standard deviation of the Gaussian filter.
- **Zoom In**: Crops the image to a square and then interpolates back to the original image shape. The only parameter is the scale factor with which to zoom in. The transform simulates capturing the image at a higher magnification level.
- **Zoom Out**: Interpolates the image to a smaller shape and then pads the edges with zeros. The only parameter is the scale factor with which to zoom out. The Zoom Out transform simulates capturing the image at a lower magnification level.
- **HED Stain** A commonly used staining augmentation proposed by Tellez et al [52]. The parameters are the staining intensity α and the staining colour β. The HED transform simulates different staining protocols.

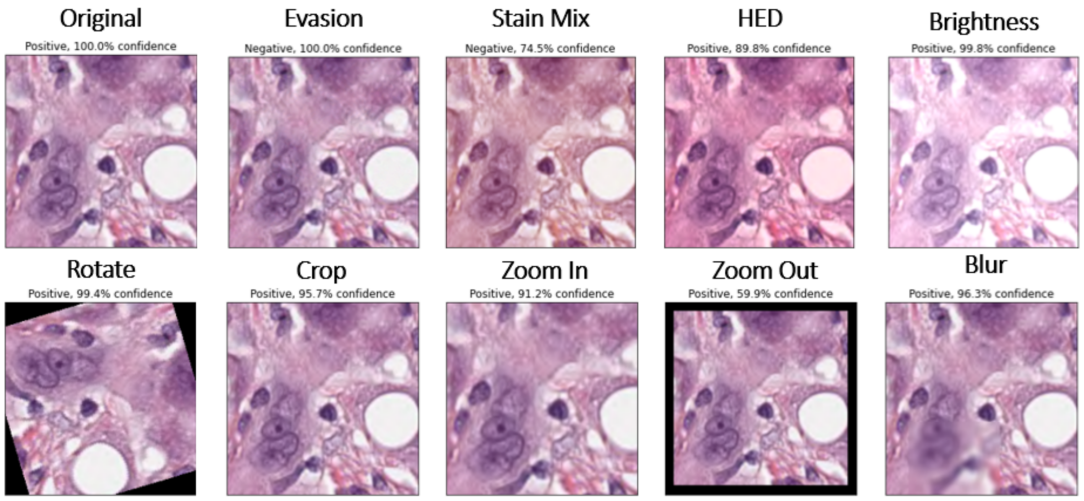

Figure 3.1: This figure shows an example of each transform in REET applied to the original image in the top left. This image is courtesy of Alex Foote. The Pixel transform is named 'Evasion' in this figure.

Chapter 4

Experimental Framework

This section will detail the framework we used to perform our experiments. We aimed to evaluate the robustness of CPath semantic segmentation and nucleus instance segmentation models. We did this by optimising the transforms in REET to simulate the most challenging environments in digital pathology. We then evaluated how well the model performed on the transformed data.

We selected REET [53] as the foundation for our experiments for several reasons. Firstly, it includes general transforms like rotations, crops, and brightness adjustments. Secondly, it includes CPath-specific transforms, such as HED and Stain Mixing, for assessing CPath model robustness. Lastly, it contains differentiable pixel-based transforms for evaluating model robustness against adversarial attacks. Most importantly, the other toolboxes we could have used lack CPath-specific transforms.

4.1 Improved Transforms

We will first discuss how we improved the transforms in REET.

The main limitation of REET is that some geometric transforms add zero padding to the image, see Rotate and Zoom Out in Figure 3.1. This is because REET takes input images at the same shape as the model's input shape. To demonstrate this, consider rotating a 270×270 image by 45°. The diagonal length of the 270×270 image is $381.84 = \sqrt{270^2 + 270^2}$. Therefore, when the image is rotated 45°, we get a rhombus with a height and width of 381.84 and a diagonal distance of 270 between the parallel edges. CNNs require a specific input shape, 270×270 in this case, so to input the image into the model, the rhombus must fit into a 270×270 square. However, the 270 diagonal distance between the parallel sides of the rhombus is insufficient to cover the 381.84 diagonal of the 270×270 square. To compensate, the rotate transform pads the uncovered area with zeros, as demonstrated in Figure 4.1.

The padding is a concern because we wish to evaluate the model's ability to classify rotated content, not its ability to recognise blank space as background. Furthermore, we plan to simulate the most challenging rotation by optimising the Rotate transform. Suppose the model is particularly good at classifying padding as background, then the optimisation may be biased towards finding the rotation angle that produces the least

padding. A rotation with minimal padding would be a rotation of 0°. Clearly, the model is no longer being evaluated on its ability to handle rotated images.

To address this, we have changed the transforms. The transforms now take images much larger than the model input shape. The transform is applied to the larger image and then cropped to the model input shape; see Figure 4.2. In this way, zero padding is not added to the image, and we can use these transforms to exclusively evaluate the conditions they simulate.

As the goal is to solely assess the model's performance on rotated images, zooming to eliminate zero padding would not be an acceptable approach.

4.2 Models

4.2.1 U-Net

We used U-Net [32] to assess the robustness of semantic segmentation models. We used a modified version of U-Net, which includes added batch normalisation layers for improved performance.

U-Net is a widely-used semantic segmentation CNN, and many segmentation models are based on its architecture [64]. Consequently, the robustness of U-Net is likely indicative of the robustness of a significant number of segmentation models used in CPath. By evaluating U-Net's robustness, we can gain insights into the performance of a broad range of models used in CPath.

The U-Net architecture we used has been taken directly from [65]. This U-Net takes input images of shape $256 \times 256 \times 3$ and outputs a single type mask of shape 256×256. See Figure 4.3 for the original U-Net architecture.

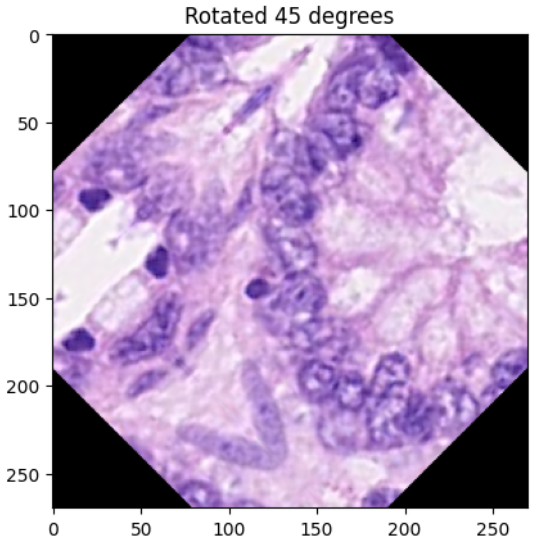

Figure 4.1: This figure shows a 270×270 image rotated 45° by REET.

4.2.2 HoVer-Net

We used HoVer-Net [33] to assess the robustness of both nucleus instance segmentation and semantic segmentation models. HoVer-Net is a highly-regarded CNN and was the best-performing nucleus instance segmentation model in the 2020 MoNuSAC challenge [66]. Evaluating the robustness of top-performing models is essential, as they will be the first models integrated into clinical practice.

HoVer-Net has three output branches: *np*, *hv*, and *tp*. The *np* branch generates a binary mask that classifies image pixels into two classes: containing nuclei or not containing nuclei. The *hv* branch produces two masks: a horizontal mask and a vertical mask. The horizontal mask indicates the horizontal distance of each nucleus pixel from its nucleus centroid. The vertical mask works analogously for the vertical direction. The *tp* branch outputs a type mask. To see the architecture of HoVer-Net, see Figure 4.4.

HoVer-Net uses highly non-linear post-processing on the output of the *np* and *hv* branches to generate an instance map. The segmentation of the instance map is then used to improve the *tp* type map. Figure 4.5 shows the order of post-processing steps.

HoVer-Net takes input images of shape $270 \times 270 \times 3$ and outputs segmentation masks of shape 80×80. The output masks give the instance segmentation and semantic segmentation information for the centre $80 \times 80 \times 3$ square of the input image.

We used the implementation of HoVer-Net found at [65] for our experiments.

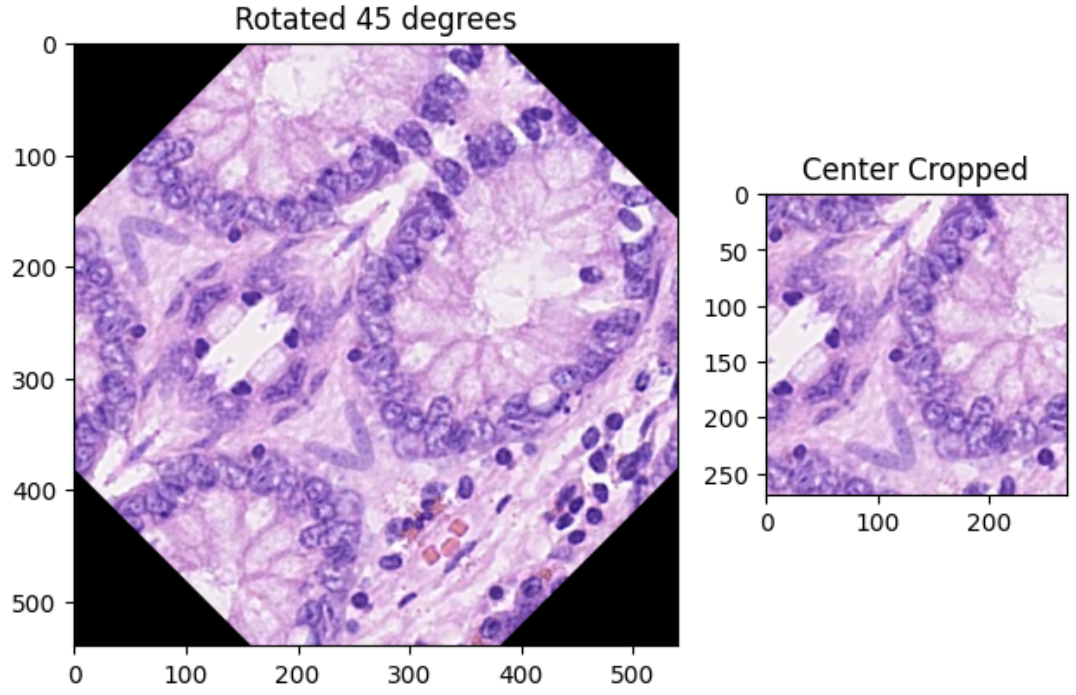

Figure 4.2: This method shows how we addressed the zero padding limitation in REET. A larger image is transformed on the left. The image on the right shows the transformed image cropped to the correct size.

4.3 Data

To train and evaluate the models, we used the CoNSeP dataset [33]. CoNSep consists of 41 H&E stained image tiles of shape 1000×1000 at 40× objective magnification. The

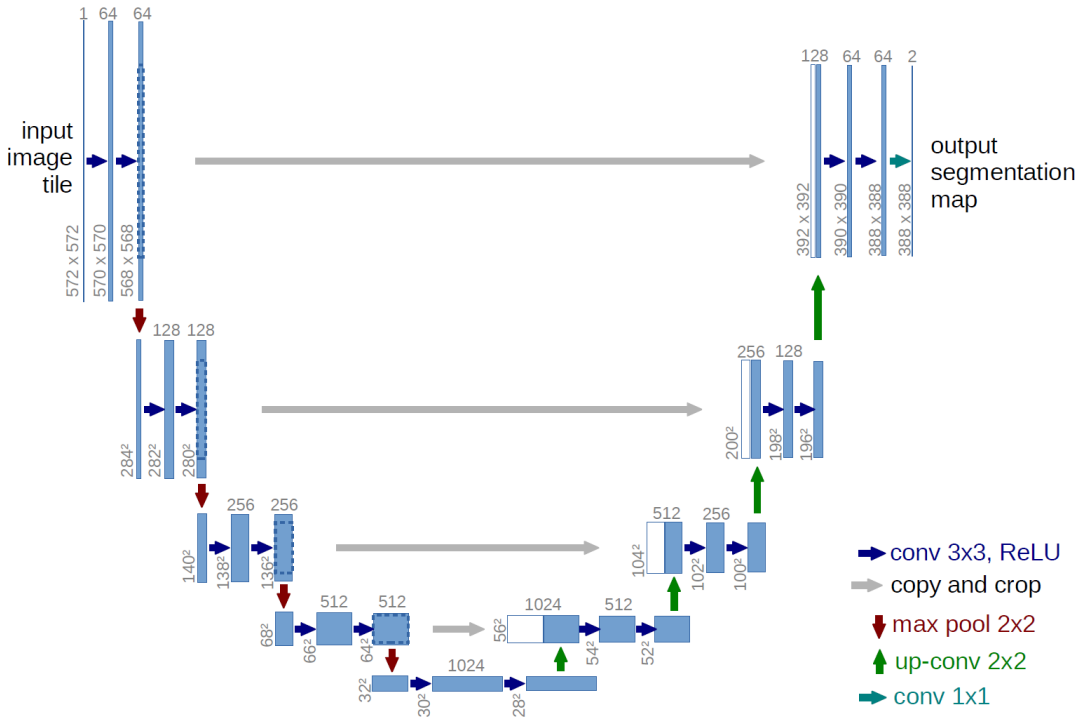

Figure 4.3: This figure shows the architecture of the original U-Net. This image was taken from [32].

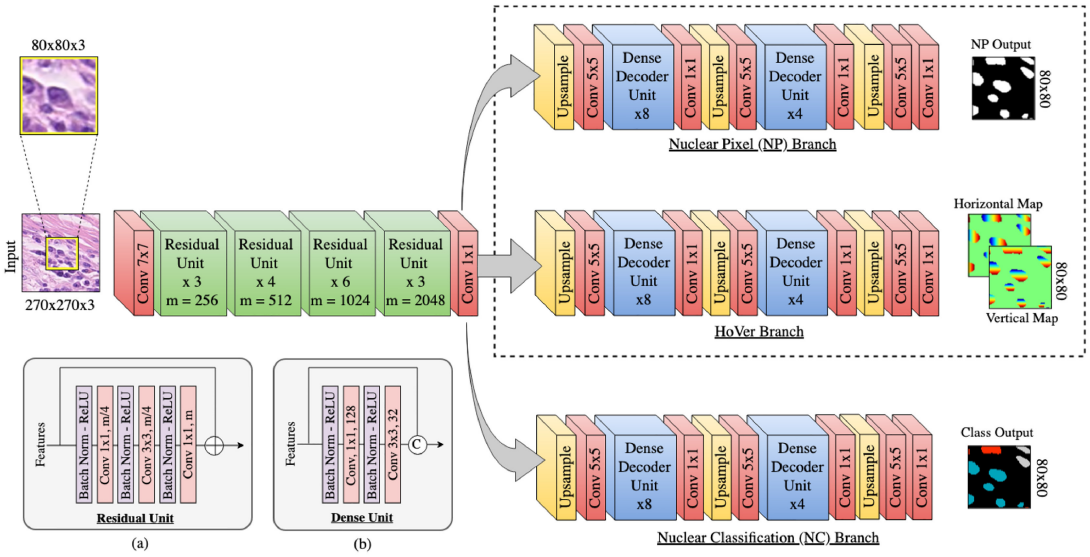

Figure 4.4: This figure shows the architecture of HoVer-Net. We use *np* to denote the top branch, *hv* to denote the middle branch and *tp* to denote the bottom branch. This image was taken from [33].

images were extracted from 16 colorectal adenocarcinoma WSIs belonging to a single patient. Each image comes with a nuclear instance map and a type map showing the ground truth values. Two expert pathologists annotated the images. See Figure 4.6 for an example of the information provided for a single image.

The dataset is separated into 5 distinct classes:

1. Background
2. Other nuclei
3. Inflammatory nuclei
4. Epithelial nuclei
5. Spindle-shaped nuclei

We divided the data into three distinct groups: a training set, a validation set, and

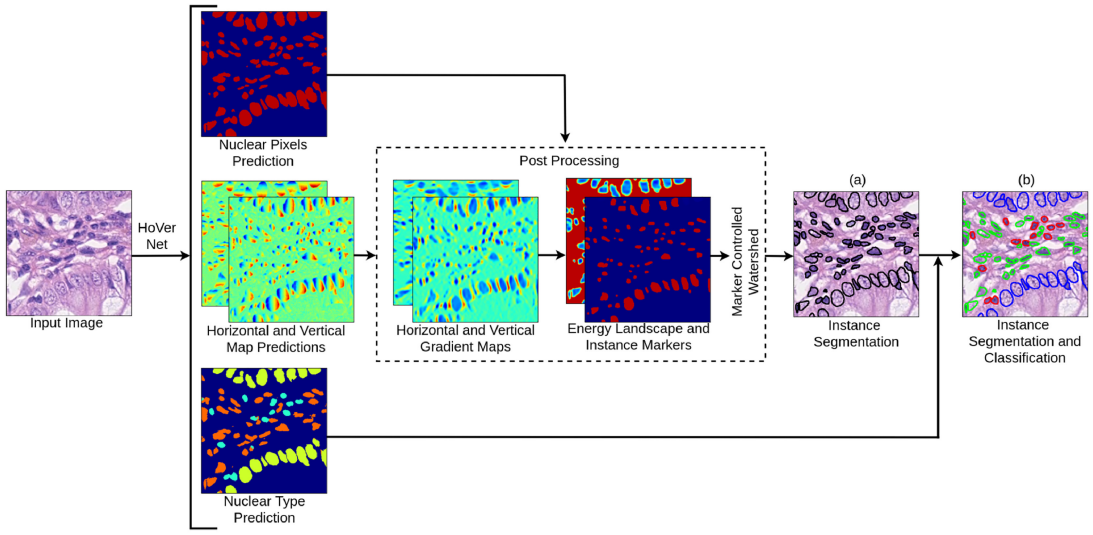

Figure 4.5: This figure shows how HoVer-Net uses the output masks to perform nucleus instance and semantic segmentation. This image was taken from [33].

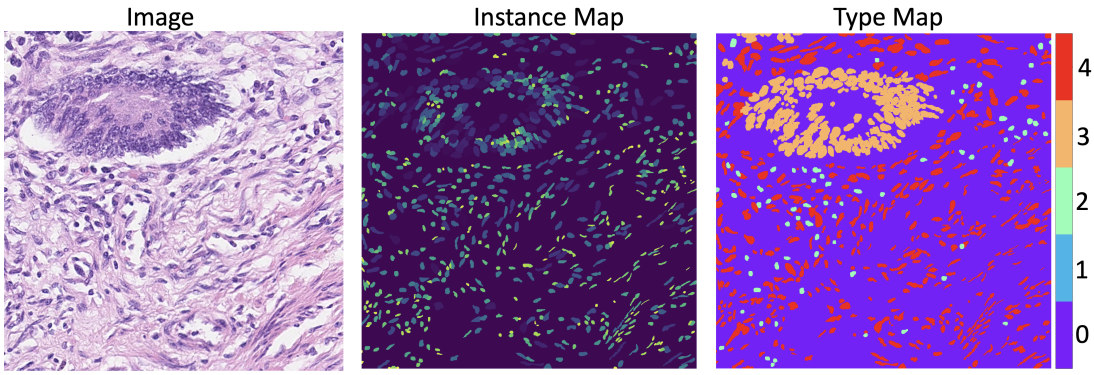

Figure 4.6: This figure shows an example of a 1000×1000 image from CoNSeP and its associated instance and type map. The type map classes are 0=Background, 1=Other nuclei, 2=Inflammatory, 3=Epithelial, 4=Spindle-shaped.

a testing set. Each image was randomly allocated to one of these groups, resulting in 29 images in the training set and 6 images each in the validation and testing sets. We then used 'extract_patches.py' from the official PyTorch implementation of HoVer-Net [65] to extract 49 mirrored 540×540 image patches for each 1000×1000 image. This mirroring technique enables us to generate more data and has been proven to improve the performance of CPath models [67].

We chose to work with this dataset because it exemplifies a key challenge in CPath that we aim to tackle: the scarcity of varied annotated data. The entire dataset comes from a single patient. We must evaluate the robustness of models trained on limited datasets before using limited datasets to train models for clinical applications.

4.4 Optimising Transforms for U-Net

To optimise the transforms for semantic segmentation, we applied untargeted PGD and untargeted stochastic search. These optimisation methods are "untargeted" because they aim to maximise pixel misclassification, irrespective of the predicted class. We used the cross-entropy loss to measure the similarity between the predicted type masks and the ground truth masks. When the similarity between the masks decreases, the cross-entropy loss increases. We aim to select a transform that maximises the loss, forcing the model to predict a type mask that is as dissimilar to the truth mask as possible. Therefore, the algorithm finds the most challenging transform for the current model on the current image.

4.4.1 Differentiable Transforms

We optimised the differentiable transforms using an adapted PGD. Let L_θ denote the cross-entropy loss with respect to model parameters θ, let y denote the truth mask, let I denote the number of iterations, let $S^{(i)}$ be the set of transform weights for transform $T_{S^{(i)}}$, let $\mathcal{O}_\epsilon$ be the optimiser with step size ϵ, let δ_α be the max perturbation size for weight α and let x_{adv} be the algorithm's output image. For our case, we assume that the optimiser returns updated weights. Then we optimise the differentiable transforms using the following:

1. Initialise transform parameters $S^{(0)}$ at random s.t. $\forall \alpha \in S^{(0)}$ we have $\|\alpha\|_p \leq \delta_\alpha$
2. $S^{(i+1)} = \text{argmin}_z \|z - \mathcal{O}_\epsilon(S^{(i)}, \nabla_{T_{S^{(i)}}}(-L_\theta(T_{S^{(i)}}(x), y)))\|_p$ s.t. $\forall \alpha \in z$, $\|\alpha\|_p \leq \delta_\alpha$
3. $n = \text{argmax}_i\{L_\theta(T_{S^{(i)}}(x), y)\}$
4. $x_{adv} = T_{S^{(n)}}(x)$

The differentiable transforms are non-geometric, so the truth masks are unchanged by the transforms.

4.4.2 Non-differentiable Transforms

We performed a stochastic search to optimise the non-differentiable transforms for semantic segmentation models. For non-differentiable transforms, $S^{(i)}$ denotes the set of parameters for transform $T_{S^{(i)}}$ and we use $(\delta_{\alpha_1}, \delta_{\alpha_2})$ to denote the selected range of

values that parameter α can take. We optimise the non-differentiable transforms using the following:

1. For each $1 \leq i \leq I$, randomly select the parameters of $S^{(i)}$ s.t. $\forall \alpha \in S^{(i)}$ we have $\delta_{\alpha_1} \leq \alpha \leq \delta_{\alpha_2}$
2. $n = \text{argmax}_i\{L_\theta(T_{S^{(i)}}(x), T'_{S^{(i)}}(y))\}$
3. $x_{adv} = T_{S^{(n)}}(x)$

Some non-differentiable transforms are geometric, so we must transform the truth masks too. We used T' to denote the appropriate transform that must be applied to the truth masks when transform T is applied to the image.

4.5 Optimising Transforms for HoVer-Net

Because the post-processing for HoVer-Net is highly non-linear, we cannot optimise transforms for HoVer-Net using the instance maps directly. We used the truth masks for each output branch to optimise the transforms. The loss $\mathcal{H}$ from the original HoVer-Net paper [33] is used to measure the similarity between the truth masks and the predicted masks at all branches. The loss $\mathcal{H}$ is defined as

$$\mathcal{H} = \underbrace{\mathcal{L}_a + \mathcal{L}_b}_{hv} + \underbrace{\mathcal{L}_c + \mathcal{L}_d}_{np} + \underbrace{\mathcal{L}_e + \mathcal{L}_f}_{tp}$$

Where $\mathcal{L}_a$ and $\mathcal{L}_b$ are the losses of the hv branch, $\mathcal{L}_c$ and $\mathcal{L}_d$ are the losses of the np branch and $\mathcal{L}_e$ and $\mathcal{L}_f$ are the losses of the tp branch. We use $\mathcal{L}_a$ to denote the mean square error of the predicted horizontal and vertical maps. We use $\mathcal{L}_b$ to denote the mean squared error of the horizontal and vertical gradients of the horizontal and vertical maps. We use $\mathcal{L}_c$ and $\mathcal{L}_e$ to denote the cross-entropy loss of the np and tp branches respectively. We use $\mathcal{L}_d$ and $\mathcal{L}_f$ to denote the dice loss of the np and tp branches respectively. For more details on how $\mathcal{L}_a$ and $\mathcal{L}_b$ are calculated, see [33].

Our framework for optimising the transforms can be summarised as maximising the loss function $\mathcal{H}$. Maximising $\mathcal{H}$ produces, for each output branch, predicted masks that are as dissimilar to the truth masks as possible. Therefore, the algorithm finds the most challenging transform for the current model on the current image.

4.5.1 Differentiable Transforms

To optimise the differentiable transforms for HoVer-Net, we used an adapted PGD. Let $\mathcal{H}_\theta$ be the loss with model parameters θ, and let y_{np}, y_{hv} and y_{tp} be the truth masks for the np, hv, and tp branches respectively. Then we optimise the transforms for HoVer-Net using the following:

1. Initialise transform parameters $S^{(0)}$ at random s.t. $\forall \alpha \in S^{(0)}$, $\|\alpha\|_p \leq \delta_\alpha$
2. $S^{(i+1)} = \text{argmin}_z \|z - \mathcal{O}_\epsilon(S^{(i)}, \nabla_{T_{S^{(i)}}}(-\mathcal{H}_\theta(T_{S^{(i)}}(x), [y_{np}, y_{hv}, y_{tp}])))\|_p$ s.t. $\forall \alpha \in z$, $\|\alpha\|_p \leq \delta_\alpha$
3. $n = \text{argmax}_i\{\mathcal{H}_\theta(T_{S^{(i)}}(x), [y_{np}, y_{hv}, y_{tp}])\}$
4. $x_{adv} = T_{S^{(n)}}(x)$

4.5.2 Non-Differentiable Transforms

To optimise the non-differentiable transforms for HoVer-Net, we used a stochastic search in the following way:

1. For each $1 \leq i \leq I$, randomly select the parameters of $S^{(i)}$ s.t. $\forall \alpha \in S^{(i)}$ we have $\delta_{\alpha_1} \leq \alpha \leq \delta_{\alpha_2}$
2. $n = \text{argmax}_i\{\mathcal{H}_\theta(T_{S^{(i)}}(x), [y_{np}, g(T'_{S^{(i)}}(y)), y_{tp}])\}$
3. $x_{adv} = T_{S^{(n)}}(x)$

As mentioned, some non-differentiable transforms are geometric, so we must also transform the truth masks. However, a geometric transform applied to the horizontal and vertical maps would mean that the maps no longer calculate horizontal and vertical distance. To address this, we use a function g to generate the new horizontal and vertical maps from the transformed instance map $T'_{S^(i)}(y)$.

4.6 Evaluating Robustness

We used the models to get predicted masks for the transformed images. We evaluated model performance by comparing the truth masks with the predicted masks. The better the models perform on the transformed images, the more robust they are.

Both HoVer-Net and U-Net produce type masks. To evaluate the quality of the type masks, we used the Dice Coefficient. Dice ranges from 0 to 1. A higher dice indicates better segmentation. We calculated Dice as follows:

$$Dice = \frac{2|TP|}{2|TP| + |FN| + |FP|}$$

Where TP are the true positives for each pixel, FN are false negatives, and FP are false positives.

We calculated Dice over the entire test dataset instead of taking the mean of the individual image patches. We cannot average over the image patches, because different patches contain different numbers of objects. Averaging over image patches can add bias if an outlier exists in an image patch with few objects.

We opted to use Dice because it is widely used in medical imaging. Therefore, our results can be easily interpreted by the research community.

We divided our evaluation of predicted type masks into two distinct cases: nuclear pixel classification and type pixel classification. Dividing the cases provides a more comprehensive understanding of the model's performance. A model may excel at segmenting nuclei from the background, but fail at nuclei type classification.

We generated a binary mask for nuclear pixel classification by bundling all nuclei classes into a single class. We used Dice to evaluate how successfully our model classifies nuclear pixels.

For type pixel classification, we calculated Dice individually for inflammatory, epithelial and spindle-shaped nuclei and took the mean of the three classes. This approach ensures that all classes of interest are considered by our metric, even those that occur

less frequently in the data. By using the mean, the metric only indicates high performance when the model effectively identifies all classes. In digital pathology, it is important that the model identifies all classes because the most important class may be sparse.

To evaluate HoVer-Net's nucleus instance segmentation, we used Panoptic Quality (PQ):

$$PQ = \frac{\sum_{(p,g)\in TP} \text{IoU}(p,g)}{|TP|} \cdot \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} = SQ \cdot RQ$$

Where TP, FP and FN indicate the true positive, false positive and false negative nuclei respectively. The predicted segment is p, the ground truth segment is g, and IoU is the intersection over union. PQ ranges from 0 to 1. A PQ of 0 means no nuclei were correctly detected. A PQ of 1 means perfect nucleus instance segmentation.

PQ can also be written as the product of segmentation quality (SQ) and recognition quality (RQ). SQ measures how well the predicted segments fit the ground truth segments, and RQ measures how well the model detects nuclei.

We calculated PQ over the entire dataset. We did this for the same reasons we calculated Dice over the entire dataset.

We chose to use PQ because it is the most widely used metric for evaluating nucleus instance segmentation. Researchers will find the results easy to compare to existing work.

4.7 Transform Parameters for Experiments

The chosen transform parameters aim to simulate realistic clinical scenarios without excessively distorting the images. The parameters used in the experiments are listed in Table 4.1. Due to time constraints, for each transform, we limited the optimisation iterations to 5. We also set $\mathcal{O}$ to be RMSProp [68] for all differentiable transforms.

Transforms	**Parameter Ranges**
Pixel	$\epsilon = 0.1$ $p = \infty$ $\delta = 5$
Stain Mixing	$\epsilon = 0.002$ $p = \infty$ $\delta = 5$
Brightness	$\epsilon = 1$ $p = \infty$ $\delta = 5$
Rotate	Angle $\in (0, 360)$
Crop	$x \in (0, 100)$ $y \in (0, 100)$ height $\in (150, 200)$ width $\in (150, 200)$
Blur	$x \in (0, 100)$ $y \in (0, 100)$ height $\in (0, 100)$ width $\in (0, 100)$ kernel size $\in (1, 10)$ $\sigma \in (1, 10)$
Zoom In	Scale $\in (1.1, 2)$
Zoom Out	Scale $\in (0.5, 0.95)$
HED Stain	$\sigma = 0.008$ $\alpha \in (1 - \sigma, 1 + \sigma)$ $\beta \in (-\sigma, +\sigma)$

Table 4.1: This table provides the transform parameter ranges used for our experiments. The parameters were carefully selected to generate realistic variations.

Chapter 5

Experiments

In this section, we describe the experiments we performed to:

1. Evaluate the robustness of HoVer-Net and U-Net.
2. Determine whether adversarial training can improve the robustness of HoVer-Net and U-Net

In our study, the method of image manipulation serves as the differentiating factor between models of the same architecture. We trained 3 U-Net models and 3 HoVer-Net models. All U-Net models used the same training parameters, and all HoVer-Net models used the same training parameters. We used three different methods of image manipulation to train the models.

We trained the U-Net models using an Adam optimiser [69], with an initial learning rate of 1×10^{-4}. We used a learning rate scheduler that reduced the learning rate by a factor of ten if the training loss did not decrease for 15 consecutive epochs. The U-Net models underwent 150 epochs of training with a batch size of 64. After the initial training, we reduced the batch size to 4, and continued training. Ultimately, we selected the models that achieved the lowest validation loss.

Similarly, we trained the HoVer-Net models using an Adam optimiser with an initial learning rate of 1×10^{-4}. We used a learning rate scheduler that reduced the learning rate by a factor of ten if the training loss did not decrease for 15 consecutive epochs. We then reduced the batch size to 4, and continued training. Ultimately, we selected the models that achieved the lowest validation loss.

All training was performed on a Nvidia A100 40GB using Google Colab.

5.1 Experiment 1: Robustness of Segmentation Models

To evaluate the robustness of segmentation models, we trained two U-Net models and two HoVer-Net models.

We trained the first set of models on the vanilla training data without augmentations. We refer to these models as *U-Net NA* and *HoVer-Net NA* to represent U-Net and

HoVer-Net trained with no augmentations.

We trained the second set of models using random augmentations. These augmentations were not optimised with respect to the current model weights. We refer to these models as *U-Net RA* and *HoVer-Net RA* to represent the models trained with random augmentations.

The random augmentations are as follows, and are applied in the following order:

1. Affine transformation with the following:
    - Scaling in x and y axes within 80-120% of the original size.
    - Translations in x and y axes within -1% to +1%.
    - Shear within -5 to +5 degrees.
    - Rotation within -179 to +179 degrees.
2. Crop the image to the model input shape.
3. Horizontal flip with a probability of 50%.
4. Vertical flip with a probability of 50%.
5. One of the following:
    - Gaussian blur with a maximum kernel size of 3.
    - Median blur with a maximum kernel size of 3.
    - Additive Gaussian noise with mean 0 and scale between 0 and 0.05×255, applied to the colour channels independently with 50% probability.
6. Sequential random order of the following:
    - Add a value to hue within the range of -8 to +8.
    - Add a value to saturation within the range of -0.2 to +0.2.
    - Add a value to brightness within the range of -26 to +26.
    - Multiply the contrast by a value within the range of 0.75 to 1.25.

This augmentation protocol has been taken from [65].

We used the experimental framework to evaluate the four models trained in this section.

5.1.1 Motivation for Experiment 1

We have made our reasons for evaluating the robustness of CPath segmentation models clear.

We evaluated models trained on un-augmented data to establish baseline levels of robustness. The baselines give the robustness of each architecture when no measures are taken to improve robustness.

Random augmentations are the most widely used method of data manipulation to improve the robustness and generalisability of CNNs. Any model deployed in the near future will be trained on randomly augmented data. Thus, by evaluating the

robustness of these models, we obtain the most accurate representation of robustness that we can expect to see in a clinical setting.

By comparing the performance of both HoVer-Net and U-Net, we can determine which architecture is more robust against clinical variations. The one that is more robust to clinical variations is better suited for clinical deployment.

5.2 Experiment 2: Adversarial Training to Improve Robustness

For this experiment, we adversarially trained HoVer-Net and U-Net.

To perform adversarial training, we applied Pixel to the training images. We applied Pixel using the algorithms and transform parameters detailed in the framework section. Before backpropagation, we applied the Pixel transform to each image in the batch. We repeated this process for each batch and each epoch.

We refer to the adversarially trained HoVer-Net and U-Net as *HoVer-Net Pix* and *U-Net Pix* to indicate that we trained them on the Pixel transform.

5.2.1 Motivation for Experiment 2

We hypothesised that adversarial training would improve robustness to the transforms in REET. If this were the case, it would imply that we can use adversarial training to improve the generalisability of CPath segmentation models.

5.3 Results

The performance of each model at nucleus pixel classification can be seen in Table 5.1 and Figure 5.1. The performance of each model at type pixel classification can be seen in Table 5.2 and Figure 5.2. The performance of each HoVer-Net model at nucleus instance segmentation can be seen in Table 5.3 and Figure 5.3.

	U-Net			HoVer-Net		
	NA	**RA**	**Pix**	**NA**	**RA**	**Pix**
Original	0.7933	0.8228	0.7523	0.7937	**0.8334**	0.7707
Pixel	0.5575	0.6106	**0.6882**	0.5279	0.5988	0.6557
Rotate	0.7771	0.8129	0.7392	0.7535	**0.8150**	0.7389
Zoom Out	0.7368	**0.7823**	0.6881	0.5986	0.7488	0.6532
Zoom In	0.7010	**0.7755**	0.6947	0.6132	0.7563	0.6773
Brightness	0.7878	0.8203	0.7444	0.7868	**0.8342**	0.7619
HED Stain	0.7520	0.7946	0.7259	0.7675	**0.8233**	0.7319
Stain Mixing	0.7089	0.7587	0.6838	0.7201	**0.8106**	0.6606
Crop	0.7308	0.7258	0.5657	0.7700	**0.8069**	0.7339
Blur	0.7321	0.7746	0.7163	0.7072	**0.8106**	0.7041

Table 5.1: This table presents the nucleus pixel classification performance on the transformed test dataset, measured using the Dice Coefficient. The NA column is the model trained on no augmentations, the RA column is the model trained on random augmentations, the Pix column is the model trained on the Pixel transform. The leftmost column specifies which transform was applied to the data. The result for the best-performing model on each transform is in bold. The "Original" row displays the results on the untransformed test data. See the Framework section for more details on how nucleus pixel classification is measured.

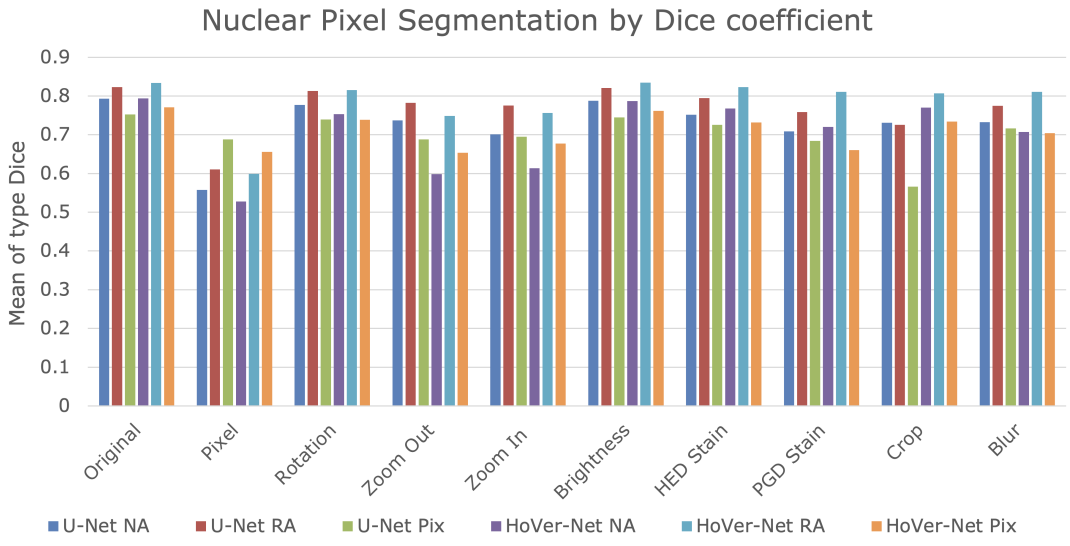

Figure 5.1: Bar chart showing the results from Table 5.1.

	U-Net			HoVer-Net		
	NA	**RA**	**Pix**	**NA**	**RA**	**Pix**
Original	0.6505	0.7078	0.6044	0.6391	**0.7356**	0.5805
Pixel	0.2901	0.3506	**0.4732**	0.2433	0.3808	0.3262
Rotate	0.6211	0.6789	0.5684	0.5390	**0.6838**	0.5014
Zoom Out	0.4169	**0.6063**	0.3694	0.3280	0.5455	0.3190
Zoom In	0.3739	**0.3741**	0.2614	0.2295	0.3281	0.2985
Brightness	0.6244	0.7009	0.5801	0.6115	**0.7306**	0.5518
HED Stain	0.5080	0.5225	0.5394	0.5597	**0.6759**	0.4853
Stain Mixing	0.5138	0.4817	0.5181	0.5255	**0.6301**	0.4425
Crop	0.5608	0.5340	0.3882	0.5265	**0.6316**	0.4854
Blur	0.5657	**0.6248**	0.5299	0.5384	0.5922	0.4885

Table 5.2: This table presents the type pixel classification performance on the transformed test dataset, measured using the Dice Coefficient. The NA column is the model trained on no augmentations, the RA column is the model trained on random augmentations, the Pix column is the model trained on the Pixel transform. The leftmost column specifies which transform was applied to the data. The result for the best-performing model on each transform is in bold. The "Original" row displays the results of the models on the untransformed test data. See the Framework section for more details on how type pixel classification is measured.

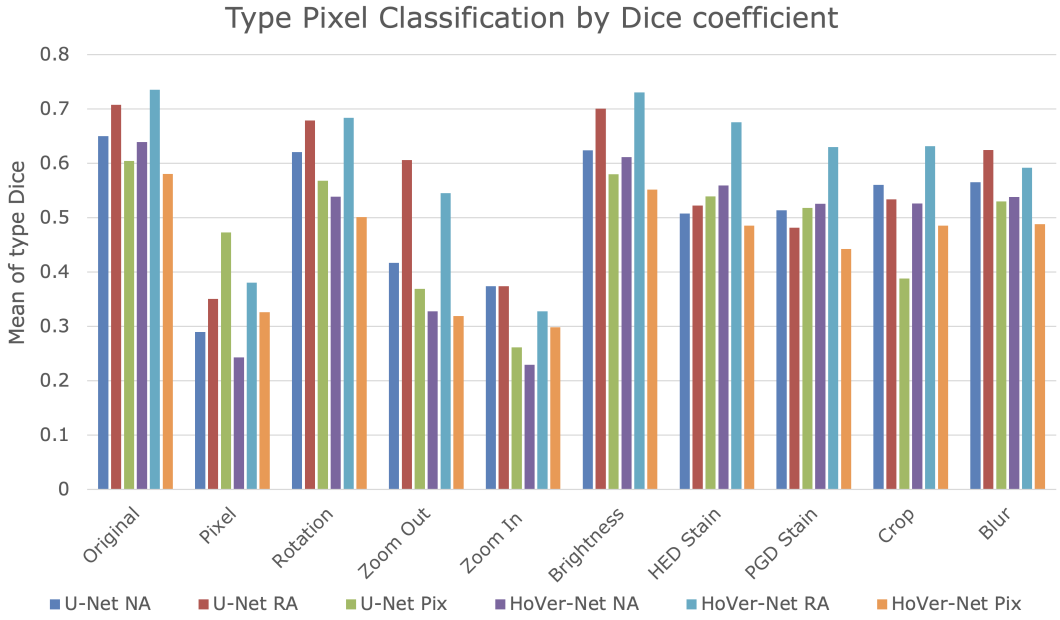

Figure 5.2: Bar chart showing the results from Table 5.2.

	HoVer-Net		
	NA	**RA**	**Pix**
Original	0.4304	**0.4716**	0.3627
Pixel	0.2159	**0.2590**	0.2477
Rotate	0.3651	**0.4316**	0.3207
Zoom Out	0.2159	**0.2905**	0.1313
Zoom In	0.2684	**0.3786**	0.2358
Brightness	0.4270	**0.4697**	0.3594
HED Stain	0.4126	**0.4614**	0.3422
Stain Mixing	0.3930	**0.4508**	0.3111
Crop	0.3846	**0.4708**	0.3055
Blur	0.3553	**0.3968**	0.3152

Table 5.3: This table presents the nucleus instance segmentation performance on the transformed test dataset, measured using the panoptic quality. The NA column is the model trained on no augmentations, the RA column is the model trained on random augmentations, the Pix column is the model trained on the Pixel transform. The leftmost column specifies which transform was applied to the data. The result for the best-performing model on each transform is in bold. The "Original" row displays the results on the untransformed test data. See the Framework section for more details on how nucleus instance segmentation performance is measured.

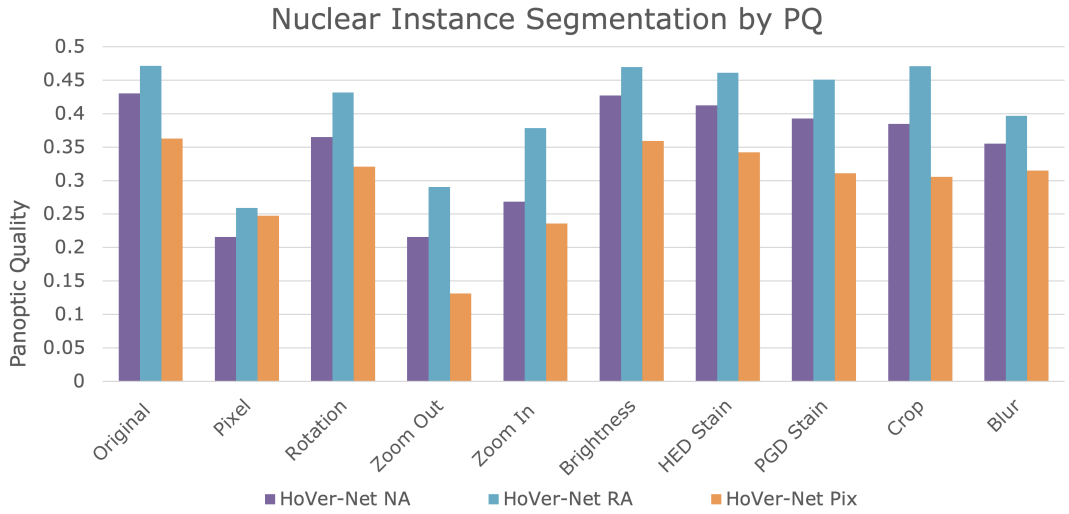

Figure 5.3: Bar chart showing the results from Table 5.3.

Chapter 6

Discussion

6.1 Experiment 1: Discussion

The results show that HoVer-Net and U-Net exhibit strong robustness to the pathology-based transforms: Rotate, Brightness, HED Stain, Stain Mixing and Blur. These transforms simulate the most probable variations in digital pathology. Consequently, pathologists can be confident that segmentation models will be robust to common clinical variations.

The results also show that augmentation training improves the robustness of HoVer-Net. We see that *HoVer-Net RA* outperformed *HoVer-Net NA* on all transforms for nucleus pixel classification, nucleus type classification and nucleus instance segmentation. The same cannot be said for U-Net. U-Net surprisingly performed worse on some transforms after augmentation training: Crop for nucleus pixel classification and Stain Mixing and Crop for type pixel classification.

The results indicate that U-Net is more robust to zooming transforms than HoVer-Net. *U-Net RA* outperformed *HoVer-Net RA* on the zooming transforms for both type classification and nuclear pixel classification.

The inferior performance of HoVer-Net on the Zoom Out transform is likely due to HoVer-Net's post-processing, which removes objects that are too small to be nuclei. In this study, we configured HoVer-Net to detect nuclei at a $40\times$ objective magnification. When we zoom out, the image is no longer at $40\times$ objective magnification. The post-processing removes nuclei from the zoomed-out image, because they now appear too small to be nuclei.

We also correctly predicted that U-Net would outperform HoVer-Net on the Zoom In transform. HoVer-Net was evaluated on 80×80 images, while U-Net was evaluated on 256×256 images. Because the images are at a $40\times$ objective magnification, zooming into an 80×80 image limits context, making segmentation difficult. In contrast, a zoomed-in 256×256 image retains more context, making the impact of the transform less pronounced. Figure 6.1 shows an example of Zoom In on an 80×80 image.

Robustness to zooming transforms may not be critical for digital pathology applications. In practice, the magnification level of a scanner can be set and maintained at the correct level for a model. It's unlikely that a clinical laboratory would experience

substantial variations in image magnification levels. As a result, robustness to zooming transforms is a less pressing concern for digital pathology.

HoVer-Net and U-Net exhibit moderate robustness to adversarial attacks; the Pixel transform is a PGD attack. Both *HoVer-Net RA* and *U-Net RA* achieve a nuclear pixel dice of approximately 0.6 on attacked images. The type classification is more severely affected, dropping from 0.7078 to 0.3506 and from 0.7356 to 0.3808 for *U-Net RA* and *HoVer-Net RA* respectively. We suspect that this is because the features used to distinguish different types of nuclei from one another are more subtle than those used to distinguish nuclei pixels from background.

Because of how severely adversarial attacks affected classification models, we expected attacked segmentation models to classify all pixels incorrectly. Our results show that this is not the case. Because segmentation and classification models perform different tasks and are measured using different metrics, it is difficult to compare their results. More research is needed, but these results suggest that segmentation models are more robust to adversarial attacks.

6.2 Experiment 2: Discussion

The results show that adversarial training does not improve the robustness of CPath models to pathology-based transforms. The models trained on the vanilla training data consistently performed better than those trained on the Pixel transform, with a few exceptions. The models trained on random augmentations also outperformed the adversarially trained models, but to a greater extent.

Because the adversarially trained models perform poorly on the transforms in REET, we suspect that adversarial training negatively impacts the generalisability of CPath

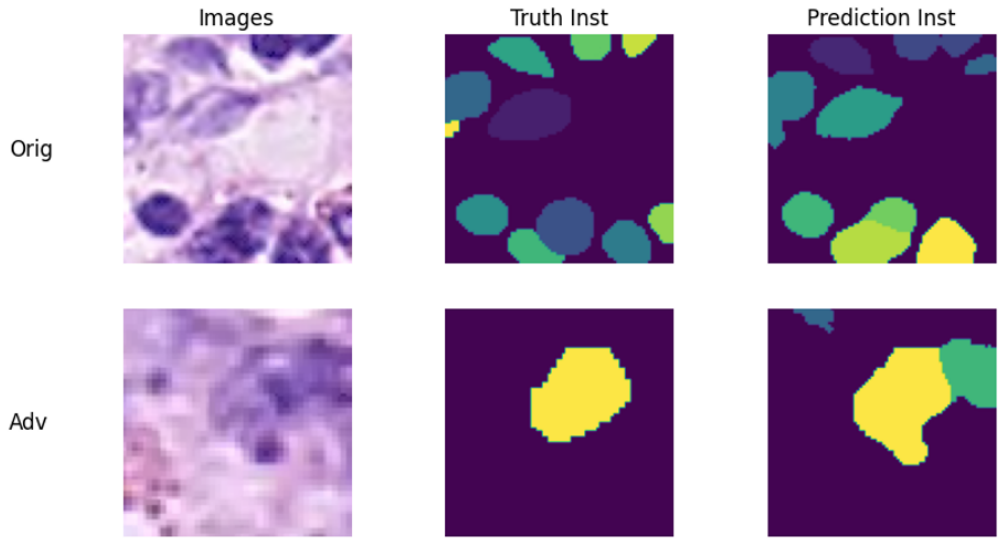

Figure 6.1: The top row shows the original image, truth instance map, and HoVer-Net's prediction. The bottom row shows the same for a zoomed image. The image was zoomed in by a scale factor of 1.81. The zoomed-in image retains little context.

models. This does not imply that robustness to adversarial attacks inherently harms generalisability, but that adversarial training does. Other methods for improving adversarial robustness exist [2]. Further research should investigate how these methods impact generalisability.

Interestingly, *HoVer-Net RA* outperforms *HoVer-Net Pix* on the Pixel transform for type classification and instance segmentation. This finding is particularly significant because adversarial training is widely regarded as the most effective way to improve the adversarial robustness of CNNs [2]. Additionally, adversarial training is considerably more computationally costly than augmentation training. If we could find an augmentation protocol that attains comparable levels of adversarial robustness to adversarial training, the implications would extend far beyond computational pathology. More research is needed.

6.3 Limitations

The results for HoVer-Net are artificially low. We are applying the post-processing to 80×80 image patches. Nuclei partially in the 80×80 image patch can appear too small to be nuclei. These nuclei are falsely removed from the prediction maps by the post-processing; see Figure 6.2. Because the 80×80 image patch is small, a large proportion of the nuclei in each patch are only partially in the image. As a result, many nuclei are falsely removed from the prediction maps when we apply post-processing to 80×80 image patches. In a clinical application, the post-processing would be applied to much larger images, such as WSIs. Thus, the results for HoVer-Net are artificially low in this study. For context, *HoVer-Net RA* had a panoptic quality of 0.5206 on the test dataset when applied directly to the 1000×1000 images. This is much higher than the 0.4716 it scored on the 80×80 image patches of the test dataset. Therefore, our results for HoVer-Net underestimate clinical performance.

In this paper, we claimed to test robustness to pathology variations by simulating the variations with augmentations. However, we did not provide empirical evidence to show that the augmentations accurately simulate these variations from the model's perspective. It is possible that the transforms do not correctly simulate the variations and may inadvertently introduce unseen features. These unseen features could be the true cause of model performance degradation. For example, when we perform a rotation, if we do not perform it by a multiple of 90°, the algorithm has to use interpolation. This interpolation adds artefacts to the image; see Figure 6.3. Rotating a tissue slide in the real world does not add these artefacts. As a result, the Rotate augmentation fails to replicate a natural rotation. If the model is particularly susceptible to these artefacts, our experiments would falsely suggest that our model struggles to deal with rotated image slides.

Another limitation is the lack of models. A sample size of two is insufficient to make scientific conclusions. We have shown that these two models are robust, but we cannot conclude that all CPath segmentation models are robust. More research is needed to evaluate the robustness of other segmentation models.

We only exposed the models to a small number of transforms. Other variations can occur in digital pathology, such as black spots and scratched glass [63]. We have not provided any evidence to suggest that our model is robust to these variations.

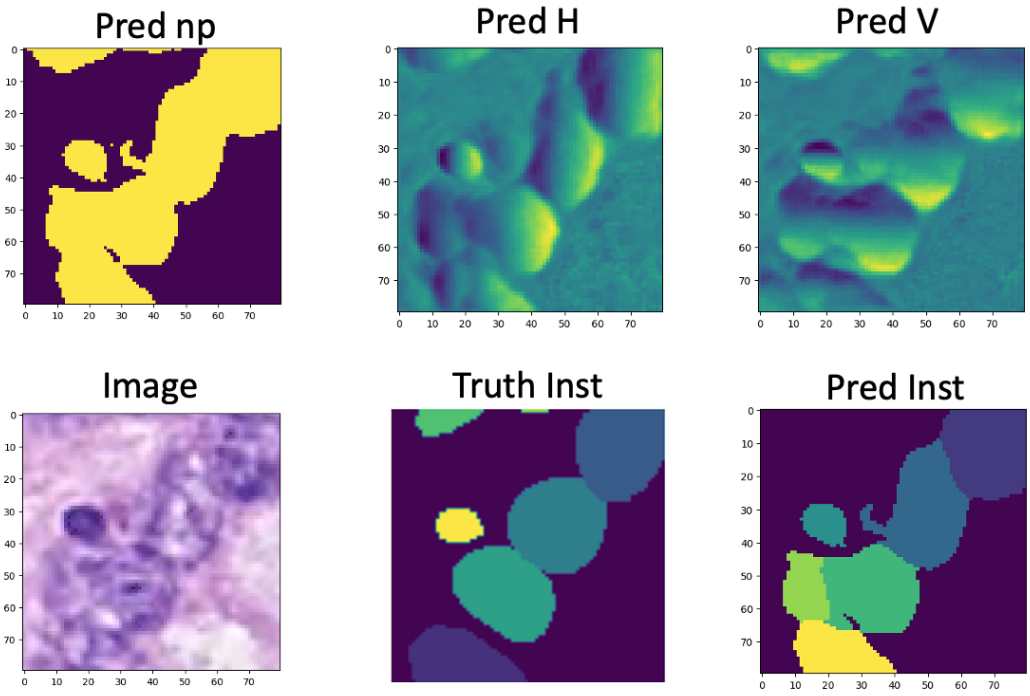

Figure 6.2: The top row shows the predicted mask from the *np* branch and the predicted horizontal and vertical maps. The second row shows the image, the truth instance map and the predicted instance map. The top left corner of the truth map shows a nucleus in the top left corner of the image. The top row shows that each output branch correctly detects this nucleus. The predicted instance map does not contain this nucleus because the post-processing determined that it was not the correct shape to be a nucleus.

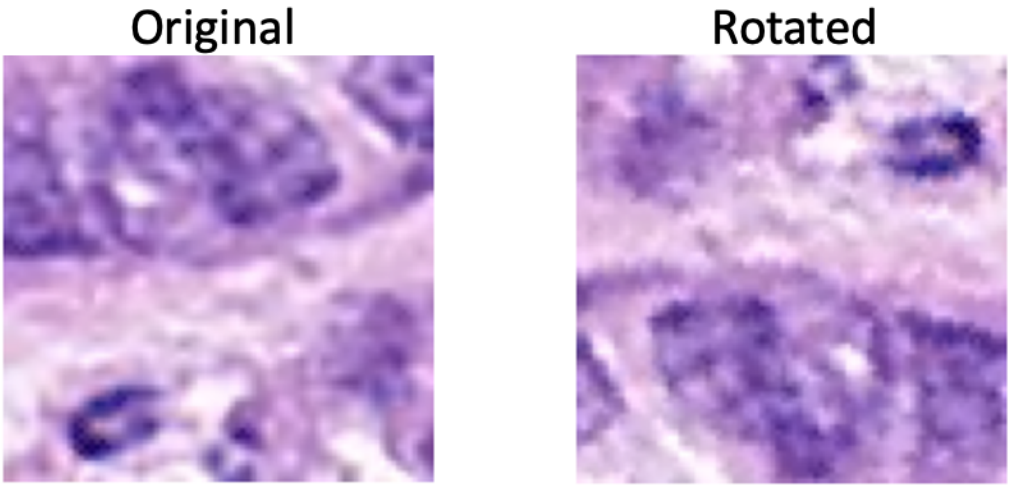

Figure 6.3: The original image is on the left. The image on the right shows the image rotated 172°. The image on the right contains small artefacts from the interpolation.

The models that we have tested are old in the context of deep learning. U-Net was released in 2015, and HoVer-Net in 2018. Newer models contain innovations that could alter robustness. We cannot assume that our results directly transfer to more modern architectures.

Chapter 7

Project Management

This section will discuss how we managed the project to complete the objectives. The project goals are listed below. Note that we did not successfully evaluate the robustness of weakly supervised CPath models on WSIs.

Original Project Goals

- ✓ 1. Evaluate the robustness of semantic segmentation models.
- ✓ 2. Evaluate the robustness of nucleus instance segmentation models.
- × 3. Evaluate the robustness of weakly supervised CPath models on WSIs

Over the next few sections, we will discuss the project's timeline in detail. The Gantt chart in Figure 7.1 shows the project's progress. The Gantt chart in Figure 7.2 shows the original, planned timeline.

7.1 Research

This project was particularly technical. Before the project started, I had no knowledge of the field. So from October to the start of December, my main objective was to develop an understanding of the literature surrounding the robustness of deep learning

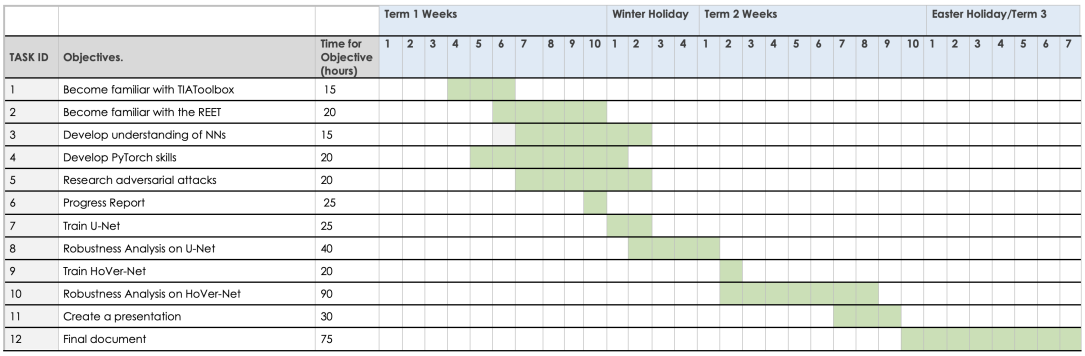

Figure 7.1: This Gantt chart shows how the project has progressed over the year.

methods and CPath. By doing this, I avoided getting stuck on problems that others had already solved.

To develop an understanding of adversarial attacks, I studied the literature reviews by Akthar et al. [2] and Xu et al. [47]. These reviews directed me to notable papers and the methods I used for my experiments. For example, these papers were the basis for my decision to use an adapted version of PGD to test robustness. Reading two distinct reviews on the robustness of deep learning mitigated the influence of author biases regarding the most effective methods.

I did this project in the Tissue Image Analytics Lab at the University of Warwick, under the supervision of CPath expert Fayyaz Minhas [70]. To gain a better understanding of CPath, Fayyaz directed me to research papers related to my project. He advised me to complete the TiaToolbox [51] tutorials to familiarise myself with CPath and suggested that I use REET as the basis for my experiments. Interactive learning with an expert was much more efficient than sifting through the papers independently.

7.2 Development

I used an agile software methodology to develop evaluation and testing tools. I used an agile methodology because it anticipates the need for flexibility [71]. The evaluation tools for semantic segmentation, nucleus instance segmentation and weakly supervised models were all separated into their own development cycles. Separating the cycles meant that I could focus on getting a functioning minimum viable product. It also meant that I could work around my deadlines for other modules by allocating sprints to periods when I was less busy, such as over the Christmas break.

The first development sprint occurred over the Christmas break. The objective of the first sprint was to implement evaluation tools for semantic segmentation. The first sprint finished on time.

The second development sprint was scheduled for week 1 to week 4 of term 2. During this sprint, I created tools to evaluate the robustness of nucleus instance segmentation models. This sprint took much longer than expected and finished in week 8 of term 2.

			Term 1 Weeks										Winter Holiday				Term 2 Weeks										Easter Holiday/Term 3						
TASK ID	Objectives.	Time for Objective (hours)	1	2	3	4	5	6	7	8	9	10	1	2	3	4	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7
1	Become familiar with TIAToolbox	15																															
2	Become familiar with the REET	20																															
3	Develop understanding of NNs	15																															
4	Develop PyTorch skills	20																															
5	Research adversarial attacks	20																															
6	Progress Report	25																															
7	Train semantic segmentation model	5																															
8	Robustness analysis of semantic segmentation	50																															
9	Train nucleus instance segmentation model	5																															
10	Robustness analysis of nucleus instance segmentation	50																															
11	Train weakly supervised WSI classification model	5																															
12	Robustness analysis study for weakly supervised WSI classification model	50																															
13	Create a presentation	25																															
14	Final document	75																															

Figure 7.2: This Gantt chart shows the timeline of the original plan for the project.

When designing attacks for nucleus instance segmentation, I planned to attack the instance map produced by HoVer-Net. However, after speaking to segmentation expert Dang Vu [72], I learned that HoVer-Net uses highly non-linear post-processing steps to generate the instance map. This means it would be impossible to track the gradients through the post-processing, which is required to perform the attack. I spent 3 weeks attempting to develop an attack for the instance map before realising it was not possible. After the 3 week delay, there was not enough time to evaluate the robustness of weakly supervised models on WSIs. Nevertheless, the agile software methodology meant that I still had viable working solutions to evaluate semantic segmentation and nucleus instance segmentation models.

The last sprint consisted of developing adversarial training tools between week 10 of term 2 and week 1 of term 3.

7.3 Experiments

We initially planned to do a much wider set of experiments. We developed tools to train on all transforms in the toolbox. We also made training tools that apply multiple transforms to a single image. We planned to train on each transform to see if any would lead to boosts in model performance. We were unable to do this because of how computationally expensive this type of training turned out to be.

For example, we designed an experiment in which the trainer randomly samples two transforms from the toolbox. The transforms are then applied to the images sequentially and optimised over 10 iterations each. To perform this single experiment on HoVer-Net with a Nvidia A100 40GB (a £10,000 GPU), it would take roughly 75 hours. Due to the limited resources available for an undergraduate project, we could not conduct multiple experiments of this magnitude. This computational constraint is the reason we limited the transform optimisation iterations to 5.

We also developed software to evaluate robustness using many metrics. We did not have enough time to input this data into tables.

7.4 Legal, Ethical and Social Concerns

Publishing these results could impact the use of AI in clinical laboratories. These results could serve as evidence that a sufficient level of robustness has been achieved for clinical integration of AI. Alternatively, these results could serve as evidence that AI has not reached a sufficient level of robustness for clinical integration. If the results are erroneous, an incorrect conclusion could be made, and lives could be lost. Therefore, before publication, we will undergo a thorough peer review process to ensure that the results are correct.

No other ethical concerns were faced in this project.

7.5 Addressing Feedback from the Presentation

Fayyaz Minhas and Dariusz Ceglarek [73] made two recommendations after the project was presented to them. The recommendations were:

1. Include a section on existing work in the field.
2. Include a section on the development methodology.

To address 1, we added the existing work section to this dissertation. The existing work section covers the following:

- Adversarial attacks.
- Augmentation training to increase the generalisability of neural networks.
- Robustness of CPath models.
- Methods used to increase robustness to adversarial attacks.
- The relationship between robustness to adversarial attacks and model generalisability.

To address 2, we included a section on how we planned the research, development and experiments for this project.

Chapter 8

Future Work

Now that we have built the tools for training and testing, we can easily extend this research to more models and datasets. Once we have tested more models, we will publish the results in a paper.

Weakly supervised models are also used in CPath [74]. These models are trained using a single label for a WSI and do not include information about the location of pathological structures. Weakly supervised models are useful for CPath because they do not require extensive labelling of WSIs. To our knowledge, no existing work has evaluated the robustness of weakly supervised models in CPath. Thus, future work could evaluate the robustness of weakly supervised models.

In clinical laboratories, CPath models will likely be used on whole slides. Therefore, we must evaluate their robustness on whole slides. One limitation of this study was that HoVer-Net's post-processing is applied to 80×80 images. This limitation means that, in this paper, HoVer-Net's performance differs from its performance in a clinic. This limitation also means that we cannot reasonably compare the robustness of HoVer-Net with U-Net, since U-Net does not have any post-processing. Future work could focus on evaluating models on whole slides, which would address the limitations and improve the quality of our results.

In this study, we trained on original images, random augmentations and the pixel transform. It would be interesting to see how training on the other transforms would affect the model's robustness. Because the other transforms in the toolbox aim to simulate real-world pathology challenges, training on these transforms would likely improve robustness and generalisability.

We also made a large number of changes to REET during this project. Others may wish to evaluate their segmentation models using our code. We intend to make our code open source. Before we can deploy the software, it will require further documentation and refactoring.

Chapter 9

Conclusion

CPath models are currently being integrated into the NHS clinical pathology workflow. Significant variations exist between clinics, posing a challenge for CPath models. Before deploying CPath models, we must empirically demonstrate their robustness to clinical variations.

Due to the shortage of diverse annotated test data, researchers have struggled to evaluate the robustness of CPath models to clinical variations. To address this, Foote et al. [30] developed REET. REET contains transforms that simulate clinical variations. The authors used the transforms to evaluate the robustness of classification models to clinical variations.

However, nuclear segmentation models are also commonly used in digital pathology. We extended the existing work by using REET transforms to evaluate the robustness of nuclear segmentation models. To test robustness, we developed algorithms to simulate the most challenging variations that CPath models can be exposed to.

We found that segmentation models exhibit strong robustness to these transforms. Consequently, we suspect that nuclear segmentation models are well-equipped to deal with clinical variations. Segmentation models appear to be significantly more robust than classification models, but more research is needed to confirm this. If segmentation models are more robust than classification models, then segmentation models are more appropriate for clinical deployment.

We also hypothesised that adversarial training would increase the robustness of nuclear segmentation models to clinical variations. To test this, we adversarially trained the models and evaluated their robustness to the optimised transforms. Our results disproved the hypothesis, showing that adversarial training negatively impacted robustness to pathology-based transforms. Based on these findings, we suspect that adversarial training negatively impacts generalisability. Therefore, using adversarially trained models for clinical deployment is not appropriate.

Interpreting these results requires caution. There is no empirical evidence to suggest that the transforms accurately simulate clinical variations from the model's perspective. We have shown that CNNs use 'invisible' features to make classifications. Our transforms may alter these features in a way that natural clinical variations do not. Further research should test the model on actual clinical variations to determine the

legitimacy of our transforms.

Our study was limited because we applied the models on individual image patches. We suspect the models will be applied to much larger images in clinical practice. The post-processing for HoVer-Net is designed to work on much larger images, so we can expect that the results for HoVer-Net underestimate true performance. Future work should focus on developing transforms for much larger images.

We used two models for our study. Because the sample size is small, we cannot conclude that these results hold for all nuclear segmentation models. In the future, now that we have developed the tools for testing nuclear segmentation models, we will extend this work to more models. Once we have done this, we will publish the results in a paper and make the code open source.

Overall, our work attempts to address a pressing challenge in pathology. Deploying non-robust models could lead to incorrect diagnoses and death. Our work suggests that pathologists looking for robust CPath models should use segmentation models.

Bibliography

[1] E. Abels, L. Pantanowitz, F. Aeffner, *et al.*, "Computational pathology definitions, best practices, and recommendations for regulatory guidance: A white paper from the Digital Pathology Association," *The Journal of Pathology*, vol. 249, no. 3, pp. 286–294, Nov. 2019, ISSN: 0022-3417. DOI: 10.1002/path.5331. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6852275/ (visited on 04/07/2023).

[2] N. Akhtar, A. Mian, N. Kardan, and M. Shah, *Advances in adversarial attacks and defenses in computer vision: A survey*, arXiv:2108.00401 [cs], Sep. 2021. [Online]. Available: http://arxiv.org/abs/2108.00401 (visited on 04/30/2023).

[3] *Pathology definition and meaning — Collins English Dictionary*, en, Apr. 2023. [Online]. Available: https://www.collinsdictionary.com/dictionary/english/pathology (visited on 04/19/2023).

[4] G. E. Abisti, *Histopathology*, en-GB. [Online]. Available: https://pathologists.org.uk/specialities/histopathology/ (visited on 04/07/2023).

[5] T. R. C. o. Pathologists, *Histopathology*. [Online]. Available: https://www.rcpath.org/discover-pathology/news/fact-sheets/histopathology.html (visited on 04/07/2023).

[6] H. Sung, J. Ferlay, R. L. Siegel, *et al.*, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," eng, *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, May 2021, ISSN: 1542-4863. DOI: 10.3322/caac.21660.

[7] CD9B0C89-877A-4819-9AFECAE124126B6F, *College report finds UK wide histopathology staff shortages*. [Online]. Available: https://www.rcpath.org/discover-pathology/news/college-report-finds-severe-staff-shortages-across-services-vital-to-cancer-diagnosis.html (visited on 04/07/2023).

[8] S. Y. Rozario, M. Sarkar, M. K. Farlie, and M. D. Lazarus, "Responding to the healthcare workforce shortage: A scoping review exploring anatomical pathologists' professional identities over time," en, *Anatomical Sciences Education*, vol. n/a, no. n/a, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ase.2260, ISSN: 1935-9780. DOI: 10.1002/ase.2260. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ase.2260 (visited on 04/08/2023).

[9] B. Märkl, L. Füzesi, R. Huss, S. Bauer, and T. Schaller, "Number of pathologists in Germany: Comparison with European countries, USA, and Canada," en, *Virchows Archiv*, vol. 478, no. 2, pp. 335–341, Feb. 2021, ISSN: 1432-2307. DOI: 10.1007/s00428-020-02894-6. [Online]. Available: https://doi.org/10.1007/s00428-020-02894-6 (visited on 04/08/2023).

[10] V. Mudenda, E. Malyangu, S. Sayed, and K. Fleming, "Addressing the shortage of pathologists in Africa: Creation of a MMed Programme in Pathology in Zambia," *African Journal of Laboratory Medicine*, vol. 9, no. 1, pp. 1–7, 2020, Publisher: AOSIS Publishing, ISSN: 2225-2010. DOI: 10.4102/ajlm.v9i1.974. [Online]. Available: http://www.scielo.org.za/scielo.php?script=sci_abstract&pid=S2225-20102020000100004&lng=en&nrm=iso&tlng=es (visited on 04/08/2023).

[11] G. Litjens, C. I. Sánchez, N. Timofeeva, *et al.*, "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis," en, *Scientific Reports*, vol. 6, no. 1, p. 26 286, May 2016, Number: 1 Publisher: Nature Publishing Group, ISSN: 2045-2322. DOI: 10.1038/srep26286. [Online]. Available: https://www.nature.com/articles/srep26286 (visited on 04/08/2023).

[12] E. A. Rakha, M. Toss, S. Shiino, *et al.*, "Current and future applications of artificial intelligence in pathology: A clinical perspective," en, *Journal of Clinical Pathology*, vol. 74, no. 7, pp. 409–414, Jul. 2021, Publisher: BMJ Publishing Group Section: Review, ISSN: 0021-9746, 1472-4146. DOI: 10.1136/jclinpath-2020-206908. [Online]. Available: https://jcp.bmj.com/content/74/7/409 (visited on 04/10/2023).

[13] L. Alzubaidi, J. Zhang, A. J. Humaidi, *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, ISSN: 2196-1115. DOI: 10.1186/s40537-021-00444-8. [Online]. Available: https://doi.org/10.1186/s40537-021-00444-8 (visited on 04/10/2023).

[14] A. Hekler, J. S. Utikal, A. H. Enk, *et al.*, "Deep learning outperformed 11 pathologists in the classification of histopathological melanoma images," en, *European Journal of Cancer*, vol. 118, pp. 91–96, Sep. 2019, ISSN: 0959-8049. DOI: 10.1016/j.ejca.2019.06.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959804919303806 (visited on 04/15/2023).

[15] Y. Liu, K. Gadepalli, M. Norouzi, *et al.*, *Detecting Cancer Metastases on Gigapixel Pathology Images*, arXiv:1703.02442 [cs], Mar. 2017. [Online]. Available: http://arxiv.org/abs/1703.02442 (visited on 04/07/2023).

[16] R. Colling, H. Pitman, K. Oien, *et al.*, "Artificial intelligence in digital pathology: A roadmap to routine use in clinical practice," en, *The Journal of Pathology*, vol. 249, no. 2, pp. 143–150, 2019, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/path.5310, ISSN: 1096-9896. DOI: 10.1002/path.5310. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/path.5310 (visited on 04/08/2023).

[17] *Pathology Group*, en-GB. [Online]. Available: https://www.ncri.org.uk/groups/pathology-group/ (visited on 04/08/2023).

[18] *About BIVDA*. [Online]. Available: https://www.bivda.org.uk/About-BIVDA (visited on 04/08/2023).

[19] *Artificial Intelligence to help save lives at five new technology centres*, en. [Online]. Available: https://www.gov.uk/government/news/artificial-

intelligence-to-help-save-lives-at-five-new-technology-centres (visited on 04/08/2023).

[20] *Funding boost for artificial intelligence in NHS to speed up diagnosis of deadly diseases*, en. [Online]. Available: https://www.gov.uk/government/news/funding-boost-for-artificial-intelligence-in-nhs-to-speed-up-diagnosis-of-deadly-diseases (visited on 04/08/2023).

[21] *About*, en-US. [Online]. Available: https://ibex-ai.com/about/ (visited on 04/10/2023).

[22] *About Us*, en-US. [Online]. Available: https://paige.ai/about-us/ (visited on 04/10/2023).

[23] *Aiforia — About Us — About Aiforia*, en. [Online]. Available: https://www.aiforia.com/about-us (visited on 04/10/2023).

[24] I. M. Analytics, *Ibex Secures PathLAKE Contracts to Roll Out AI-based Cancer Diagnostics to UK Hospitals*, en. [Online]. Available: https://www.prnewswire.com/il/news-releases/ibex-secures-pathlake-contracts-to-roll-out-ai-based-cancer-diagnostics-to-uk-hospitals-301764634.html (visited on 04/07/2023).

[25] *Paige Named Award Recipient of the PathLAKE Plus AI Project under HealthTrust Europe's National Tender for Artificial Intelligence Solutions*, en-US, Mar. 2023. [Online]. Available: https://paige.ai/paige-named-award-recipient-of-the-pathlake-plus-ai-project-under-healthtrust-europes-national-tender-for-artificial-intelligence-solutions/ (visited on 04/10/2023).

[26] *Aiforia receives a contract award to provide AI solutions for 25 NHS hospitals of the PathLAKE Plus consortium in the UK*, en. [Online]. Available: https://www.aiforia.com/blog/contract-award-nhs-hospitals-pathlake-plus-consortium (visited on 04/10/2023).

[27] S. Morales, K. Engan, and V. Naranjo, "Artificial intelligence in computational pathology – challenges and future directions," en, *Digital Signal Processing*, vol. 119, p. 103 196, Dec. 2021, ISSN: 1051-2004. DOI: 10.1016/j.dsp.2021.103196. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200421002359 (visited on 04/29/2023).

[28] A. H. Fischer, K. A. Jacobson, J. Rose, and R. Zeller, "Hematoxylin and eosin staining of tissue and cell sections," eng, *CSH protocols*, vol. 2008, pdb.prot4986, May 2008. DOI: 10.1101/pdb.prot4986.

[29] D. Tellez, G. Litjens, P. Bandi, *et al.*, "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology," *Medical Image Analysis*, vol. 58, p. 101 544, Dec. 2019, arXiv:1902.06543 [cs], ISSN: 13618415. DOI: 10.1016/j.media.2019.101544. [Online]. Available: http://arxiv.org/abs/1902.06543 (visited on 04/22/2023).

[30] A. Foote, A. Asif, N. Rajpoot, and F. Minhas, *REET: Robustness Evaluation and Enhancement Toolbox for Computational Pathology*, arXiv:2201.12311 [cs], Jan. 2022. DOI: 10.48550/arXiv.2201.12311. [Online]. Available: http://arxiv.org/abs/2201.12311 (visited on 04/07/2023).

[31] A. Zuraw, *Instance segmentation for digital pathology tissue image analysis – what is it and why do we need it?* en-US, Apr. 2021. [Online]. Available: https://digitalpathologyplace.com/instance-segmentation-for-digital-

pathology-tissue-image-analysis-what-is-it-and-why-do-we-need-it/ (visited on 04/20/2023).

[32] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv:1505.04597 [cs], May 2015. DOI: 10.48550/arXiv.1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597 (visited on 04/30/2023).

[33] S. Graham, Q. D. Vu, S. E. A. Raza, *et al.*, *HoVer-Net: Simultaneous Segmentation and Classification of Nuclei in Multi-Tissue Histology Images*, arXiv:1812.06499 [cs], Nov. 2019. [Online]. Available: http://arxiv.org/abs/1812.06499 (visited on 04/30/2023).

[34] A. Foote, A. Asif, A. Azam, T. Marshall-Cox, N. Rajpoot, and F. Minhas, *Now You See It, Now You Dont: Adversarial Vulnerabilities in Computational Pathology*, arXiv:2106.08153 [cs, eess], Jun. 2021. [Online]. Available: http://arxiv.org/abs/2106.08153 (visited on 04/10/2023).

[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards Deep Learning Models Resistant to Adversarial Attacks*, arXiv:1706.06083 [cs, stat], Sep. 2019. [Online]. Available: http://arxiv.org/abs/1706.06083 (visited on 04/07/2023).

[36] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385 [cs], Dec. 2015. DOI: 10.48550/arXiv.1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385 (visited on 04/07/2023).

[37] J. Gamper, N. A. Koohbanani, K. Benes, *et al.*, *PanNuke Dataset Extension, Insights and Baselines*, arXiv:2003.10778 [cs, eess, q-bio] version: 7, Apr. 2020. [Online]. Available: http://arxiv.org/abs/2003.10778 (visited on 12/12/2022).

[38] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, *Intriguing properties of neural networks*, arXiv:1312.6199 [cs], Feb. 2014. [Online]. Available: http://arxiv.org/abs/1312.6199 (visited on 04/11/2023).

[39] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," en, *Mathematical Programming*, vol. 45, no. 1, pp. 503–528, Aug. 1989, ISSN: 1436-4646. DOI: 10.1007/BF01589116. [Online]. Available: https://doi.org/10.1007/BF01589116 (visited on 04/11/2023).

[40] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, arXiv:1412.6572 [cs, stat], Mar. 2015. [Online]. Available: http://arxiv.org/abs/1412.6572 (visited on 04/11/2023).

[41] A. Kurakin, I. Goodfellow, and S. Bengio, *Adversarial examples in the physical world*, arXiv:1607.02533 [cs, stat], Feb. 2017. [Online]. Available: http://arxiv.org/abs/1607.02533 (visited on 04/11/2023).

[42] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Mar. 2016, pp. 372–387. DOI: 10.1109/EuroSP.2016.36.

[43] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, Oct. 2019, arXiv:1710.08864 [cs, stat], ISSN: 1089-778X, 1089-778X, 1941-0026. DOI: 10.1109/TEVC.2019.2890858. [Online]. Available: http://arxiv.org/abs/1710.08864 (visited on 04/11/2023).

[44] N. Carlini and D. Wagner, *Towards Evaluating the Robustness of Neural Networks*, arXiv:1608.04644 [cs], Mar. 2017. DOI: 10.48550/arXiv.1608.04644.

[Online]. Available: http://arxiv.org/abs/1608.04644 (visited on 04/11/2023).

[45] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, *Provably Minimally-Distorted Adversarial Examples*, arXiv:1709.10207 [cs], Feb. 2018. DOI: 10.48550/arXiv.1709.10207. [Online]. Available: http://arxiv.org/abs/1709.10207 (visited on 04/11/2023).

[46] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, *Universal adversarial perturbations*, arXiv:1610.08401 [cs, stat], Mar. 2017. DOI: 10.48550/arXiv.1610.08401. [Online]. Available: http://arxiv.org/abs/1610.08401 (visited on 04/11/2023).

[47] H. Xu, Y. Ma, H. Liu, *et al.*, *Adversarial Attacks and Defenses in Images, Graphs and Text: A Review*, arXiv:1909.08072 [cs, stat], Oct. 2019. [Online]. Available: http://arxiv.org/abs/1909.08072 (visited on 04/11/2023).

[48] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, Jul. 2019, ISSN: 2196-1115. DOI: 10.1186/s40537-019-0197-0. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0 (visited on 04/12/2023).

[49] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, *AutoAugment: Learning Augmentation Policies from Data*, arXiv:1805.09501 [cs, stat], Apr. 2019. DOI: 10.48550/arXiv.1805.09501. [Online]. Available: http://arxiv.org/abs/1805.09501 (visited on 04/12/2023).

[50] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, *Adversarial AutoAugment*, arXiv:1912.11188 [cs, stat], Dec. 2019. DOI: 10.48550/arXiv.1912.11188. [Online]. Available: http://arxiv.org/abs/1912.11188 (visited on 04/30/2023).

[51] J. Pocock, S. Graham, Q. D. Vu, *et al.*, "TIAToolbox as an end-to-end library for advanced tissue image analytics," en, *Communications Medicine*, vol. 2, no. 1, pp. 1–14, Sep. 2022, Number: 1 Publisher: Nature Publishing Group, ISSN: 2730-664X. DOI: 10.1038/s43856-022-00186-5. [Online]. Available: https://www.nature.com/articles/s43856-022-00186-5 (visited on 04/13/2023).

[52] D. Tellez, M. Balkenhol, I. Otte-Höller, *et al.*, "Whole-Slide Mitosis Detection in H&E Breast Histology Using PHH3 as a Reference to Train Distilled Stain-Invariant Convolutional Networks," *IEEE Transactions on Medical Imaging*, vol. 37, no. 9, pp. 2126–2136, Sep. 2018, Conference Name: IEEE Transactions on Medical Imaging, ISSN: 1558-254X. DOI: 10.1109/TMI.2018.2820199.

[53] A. Foote, A. Asif, N. Rajpoot, and F. Minhas, "REET: Robustness evaluation and enhancement toolbox for computational pathology," *Bioinformatics (Oxford, England)*, vol. 38, no. 12, pp. 3312–3314, May 2022, tex.eprint: https://academic.oup.com/bioinformatics/article-pdf/38/12/3312/44045223/btac315.pdf, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac315. [Online]. Available: https://doi.org/10.1093/bioinformatics/btac315.

[54] D. Stutz, M. Hein, and B. Schiele, *Disentangling Adversarial Robustness and Generalization*, arXiv:1812.00740 [cs, stat], Apr. 2019. [Online]. Available: http://arxiv.org/abs/1812.00740 (visited on 04/30/2023).

[55] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, *Robustness May Be at Odds with Accuracy*, arXiv:1805.12152 [cs, stat], Sep. 2019. [Online]. Available: http://arxiv.org/abs/1805.12152 (visited on 04/30/2023).

[56] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, *Is Robustness the Cost of Accuracy? – A Comprehensive Study on the Robustness of 18 Deep Image Classification Models*, arXiv:1808.01688 [cs], Mar. 2019. [Online]. Available: http://arxiv.org/abs/1808.01688 (visited on 04/30/2023).

[57] J. Gilmer, L. Metz, F. Faghri, *et al.*, *Adversarial Spheres*, arXiv:1801.02774 [cs], Sep. 2018. [Online]. Available: http://arxiv.org/abs/1801.02774 (visited on 04/30/2023).

[58] M.-I. Nicolae, M. Sinn, M. N. Tran, *et al.*, *Adversarial Robustness Toolbox v1.0.0*, arXiv:1807.01069 [cs, stat], Nov. 2019. DOI: 10.48550/arXiv.1807.01069. [Online]. Available: http://arxiv.org/abs/1807.01069 (visited on 04/12/2023).

[59] J. Rauber, W. Brendel, and M. Bethge, *Foolbox: A Python toolbox to benchmark the robustness of machine learning models*, arXiv:1707.04131 [cs, stat], Mar. 2018. DOI: 10.48550/arXiv.1707.04131. [Online]. Available: http://arxiv.org/abs/1707.04131 (visited on 04/12/2023).

[60] G. W. Ding, L. Wang, and X. Jin, *Advertorch v0.1: An Adversarial Robustness Toolbox based on PyTorch*, arXiv:1902.07623 [cs, stat], Feb. 2019. DOI: 10.48550/arXiv.1902.07623. [Online]. Available: http://arxiv.org/abs/1902.07623 (visited on 04/12/2023).

[61] *Robustness package*, original-date: 2019-08-21T09:26:33Z, Apr. 2023. [Online]. Available: https://github.com/MadryLab/robustness (visited on 04/12/2023).

[62] N. Ghaffari Laleh, D. Truhn, G. P. Veldhuizen, *et al.*, "Adversarial attacks and adversarial robustness in computational pathology," en, *Nature Communications*, vol. 13, no. 1, p. 5711, Sep. 2022, Number: 1 Publisher: Nature Publishing Group, ISSN: 2041-1723. DOI: 10.1038/s41467-022-33266-0. [Online]. Available: https://www.nature.com/articles/s41467-022-33266-0 (visited on 04/14/2023).

[63] B. Schömig-Markiefka, A. Pryalukhin, W. Hulla, *et al.*, "Quality control stress test for deep learning-based diagnostic model in digital pathology," en, *Modern Pathology*, vol. 34, no. 12, pp. 2098–2108, Dec. 2021, Number: 12 Publisher: Nature Publishing Group, ISSN: 1530-0285. DOI: 10.1038/s41379-021-00859-x. [Online]. Available: https://www.nature.com/articles/s41379-021-00859-x (visited on 04/12/2023).

[64] N. Siddique, P. Sidike, C. Elkin, and V. Devabhaktuni, "U-Net and its variants for medical image segmentation: Theory and applications," *IEEE Access*, vol. 9, pp. 82 031–82 057, 2021, arXiv:2011.01118 [cs, eess], ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3086020. [Online]. Available: http://arxiv.org/abs/2011.01118 (visited on 04/30/2023).

[65] *GitHub - vqdang/hover_net: Simultaneous Nuclear Instance Segmentation and Classification in H&E Histology Images.* [Online]. Available: https://github.com/vqdang/hover_net (visited on 04/30/2023).

[66] *MoNuSAC 2020 - Grand Challenge*, en. [Online]. Available: https://monusac-2020.grand-challenge.org/Results/ (visited on 04/30/2023).

[67] C. Zhang, N. Bao, H. Sun, *et al.*, "A Deep Learning Image Data Augmentation Method for Single Tumor Segmentation," *Frontiers in Oncology*, vol. 12, 2022, ISSN: 2234-943X. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fonc.2022.782988 (visited on 04/30/2023).

[68] S. Ruder, *An overview of gradient descent optimization algorithms*, arXiv:1609.04747 [cs], Jun. 2017. [Online]. Available: http://arxiv.org/abs/1609.04747 (visited on 05/01/2023).

[69] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017. DOI: 10.48550/arXiv.1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 05/06/2023).

[70] *Fayyaz Minhas*. [Online]. Available: https://warwick.ac.uk/fac/sci/dcs/people/fayyaz_minhas/ (visited on 04/22/2023).

[71] *What is Agile Software Development (Agile Methodologies)?* en. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development (visited on 04/13/2023).

[72] *Dang Vu*. [Online]. Available: https://warwick.ac.uk/study/csde/gsp/eportfolio/directory/pg/u1983886/ (visited on 04/22/2023).

[73] *WMG :: Our People :: Profile*. [Online]. Available: https://warwick.ac.uk/fac/sci/wmg/people/profile/?wmgid=462 (visited on 04/29/2023).

[74] S. Graham, F. Minhas, M. Bilal, *et al.*, *Screening of normal endoscopic large bowel biopsies with artificial intelligence: A retrospective study*, en, Pages: 2022.10.17.22279804, Oct. 2022. DOI: 10.1101/2022.10.17.22279804. [Online]. Available: https://www.medrxiv.org/content/10.1101/2022.10.17.22279804v3 (visited on 04/30/2023).