Assessing the Impact of Stain Normalization on a Cell Classification Model in Digital Histopathology

Albert Aillet and Filip Frisk

***Abstract*—In the field of digital histopathology, computer-aided diagnosis of digitized tissue samples with computational algorithms is a rising research field. The tissue samples in this study are stained using chemicals that enhance the recognizability of different tissue structures. This staining can be highly variable, which has an impact on the performance of the computational algorithms. The aim of this project is to assess the use of three color normalization algorithms as a pre-processing step on the KI dataset from a collaborative research project between Karolinska Institutet and KTH Royal Institute of Technology. The color normalization algorithms aim to reduce the color variability of the data. The basis of the study is an implementation of the EfficentNet Convolutional Neural Network classification model, that was adapted for the specific needs of the study. Performance was assessed by firstly applying the color normalization filters to the dataset and training multiple models on each of the filtered datasets. The results from the individually trained models and the combined results with ensemble learning techniques were then analyzed. Our conclusions are clear, stain normalization filters significantly impacts classification performance metrics. The impact depends on the staining qualities of the filters. Ensemble learning techniques present a more robust performance than the individual filters with a performance comparable to the best performing filter.**

***Sammanfattning*—Datorstödd diagnos av digitaliserade vävnadsprov med hjälp av beräkningsalgoritmer inom digital histopatologi är ett aktivt forskningsfält. Vävnadsproven i denna studie har färgats med kemikalier som förbättrar igenkännandet av olika vävnadsstrukturer. Kvaliteten på denna färgningsprocess kan variera, vilket har en inverkan på beräkningsalgoritmernas prestanda. Syftet med detta projekt är att utvärdera användningen av tre färgnormaliseringsalgoritmer som ett förbehandlingssteg på ett dataset från ett samarbetsprojekt mellan Karolinska Institutet och Kungliga Tekniska Högskolan. De använda färgnormaliseringsalgoritmerna syftar till att minska färgvariabiliteten i datan. Grund för studien är en implementering av klassificeringsmodellen EfficentNet, som anpassades utifrån studiens specifika behov. Prestandan bedömdes genom att först använda varje färgnormaliseringsalgoritm på datasetet och träna flera modeller på var och en av de filtrerade dataseten. Därefter analyserades resultaten från de individuella modellerna och de kombinerade resultaten med ”ensemble learning”-tekniker. Våra slutsatser är tydliga, färgnormaliseringen påverkar significant prestandamätvärdena. Dess inverkan beror på filtrens färgningsegenskaper. ”Ensemble learning” teknikerna ger en mer robust prestanda än de enskilt tränade modellerna som lika bra som det bäst presterande filtret.**

***Index Terms*—Digital pathology, Machine learning, Color normalization**

Supervisors: Rachael Sugars & Karl Meinke

TRITA number: TRITA-EECS-EX-2021:198

I. Introduction

Artificial intelligence and machine learning approaches, specifically deep learning models are part of a rising research field within digital healthcare, especially digital histopathology. The workflows of pathologists have in the past been limited to physical samples and analog microscopes. Recent developments in hardware and software have led to a digitization of this workflow. This opens up for the use of deep learning to provide pathologists with reliable support for diagnostic assessment and treatment decisions [1], [2].

Studies have shown that the need for pathology services is high, especially in low to medium income countries. Such countries have more than average disease cases but a low share of global healthcare resources and poor access to quality pathology and laboratory medicine [3]. Even in western countries the access is not evenly distributed across regions and some severely lack competence [4]. This puts heavy load on the available specialists and creates long waiting times in an already pressured healthcare system.

Since 2018, the Oral Biology and Medicine Group at the Department of Dental Medicine at Karolinska Institutet (KI) and the Theoretical Computer Science Division (EECS school) at KTH Royal Institute of Technology school have an ongoing research project on this topic named *Evaluation of Neural Networks for Digital Pathology on High Performance GPUs*. The overarching aim of the project is to provide clinicians with computer-aided diagnostic support.

Prior to this project, multiple Masters and PhD students have been involved, a dataset has been created and different machine learning techniques have been evaluated. The KI dataset consists of cell types from oral mucosa tissue samples hand-labeled by pathologists at the Department of Dental Medicine at KI. At first two deep learning algorithms, Softmax CNN and RCCNet were investigated and the results were not satisfactory in terms of accuracy [5]. A more computationally intensive deep neural network EfficientNet has been used and trained on the Kebnekaise supercomputer [6] in the Masters thesis *Epithelial Layer Boundary Detection using Graph Convolutional Networks for Digital Pathology* [7]. In this thesis, it was proposed that the color variability from the staining process could explain the misclassifications and weak generalizability. This study aimed to investigate color variability, by using methods from the study of Pontalba et al. [8].

Pontalba et al. [8] found that approaches of combining multiple color normalization filters and using ensemble learning techniques might address some of the problems associated

with color variability for a segmentation task. Similarly to the Pontalba et al. this bachelor thesis investigated the use of color normalization filters as a pre-processing step but for a cell classification model instead of a segmentation model. The impact on performance for models trained on the color normalized datasets was analyzed individually and the results from the individual models were combined using ensemble learning techniques.

II. Background

A. Histopathology

1) General: Histopathology is a field of clinical medicine where diagnosis is based on visual examination by pathologists of tissue samples under a microscope. The visual review of a tissue is often subjective, with great variability in the decision depending on the pathologist and the lab. Manual examination of samples is a laborious and time consuming task, especially if the few field specialists that are already in high demand are required [9].

2) Digital Histopathology: The recent development of the digitization of histological samples has enabled a large number of samples to be scanned and archived digitally. A common process is whole slide imaging (WSI) where tissue samples placed on glass slides are digitally scanned [10]. Digital histopathology encompasses all technologies that use these digital slides to allow for improvements and innovations in the workflow of pathologists [11]. Computational algorithms or more specifically AI algorithms can take advantage of the datasets consisting of tissue samples available for analysis to support the pathologists in the diagnosis process [12]. While pathologists have to take the final decision, the AI can highlight structures of interest in the tissue samples. However, these samples need to be annotated by experts to be of use for the AI algorithms which is a long and time consuming task. Consequently the field suffers from a lack of quality annotated data [13]. Construction of an end-to-end WSI deep learning analysis pipeline that can be used in a clinical setting requires many steps, see Fig. 1.

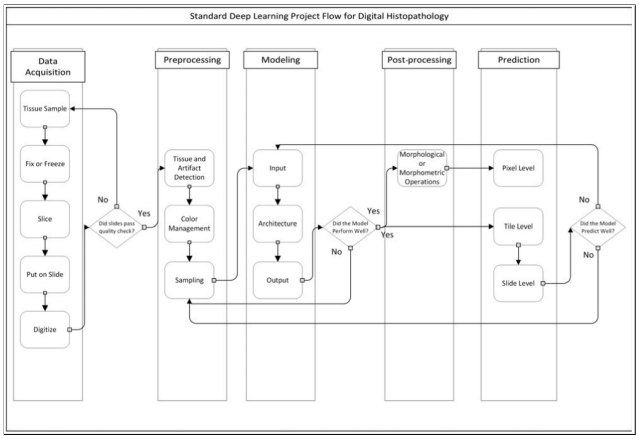

Fig. 1. Typical deep learning project flow in digital histopathology [2]

3) Clinical samples and associated histological grading: The digitized tissue samples in the study *Histopathological Grading of Oral Mucosal Chronic Graft-versus-Host Disease: Large Cohort Analysis* [14] have been histologically graded (G0 to G4). The grading refers to a points-based grading system based on ”intraepithelial lymphocytes and band-like inflammatory infiltrate, atrophic epithelium with basal cell liquefaction degeneration, including apoptosis, as well as separation of epithelium and pseudo-rete ridges” [14]. The grading gives an indication of the histological severity of the tissue sample.

4) Haematoxylin and eosin tissue sample staining and digitization process: Before a tissue sample can be scanned and digitized or observed directly by a pathologist, it has to go through a number of preparation steps to preserve its structure and have an appearance that facilitates the diagnosis of the pathologist [12]. One of the main steps is the staining process. After the initial processing, most tissues and cells are transparent under the microscopy [15] and staining is used to reveal the anatomical features of the tissue structure for visual examination. One of the most common staining processes is Haematoxylin and Eosin (H&E) staining [16]. Eosin is acidic and negatively charged and stains structures like the cytoplasm and extracellular matrix in a red or pink color. Haematoxylin is basic and stains structures like the nuclei in a purple or blue color [17]. After the cut section have been exposed to these two stains they present visually recognizable features that are easier for the pathologist to identify. However, this staining is highly variable and can produce largely different colors depending on a multitude of factors such as different staining times, the variable concentration and pH of the staining solutions [12] or the stain suppliers. In the review article *the haematoxylin and eosin stain in anatomic pathology* [18] Mark R. Wick presents some of the specific problems that can occur during staining that cause variability in the quality of the sections. The irregular staining of the sections, a poor definition between the nuclei and the cytoplasm, an over- or understaining with either of the stains or a blue-black precipitate in the stained sections are some features that contribute to a low quality section.

The tissue samples that are analyzed in this project are sampled using a 5mm punch biopsy from the oral mucosa which is the mucous membrane of the inside of the mouth. The sample is then fixed in a paraformaldehyde solution to minimize the breakdown of the tissue structure before being dehydrated and embedded in paraffin wax. The paraffin embedded samples are then sliced into thin sections, placed on glass slides (often using a microtome [12]). These slices are then deparaffinised and rehydrated and the formerly described H&E staining is applied [5], [18]. After this the section is analyzed under a microscope or digitized with a scanner and analyzed on a computer screen. This digitization process can also introduce variabilities in the digitized tissue samples depending on the use of different digitization systems [12].

As these variabilities can have great consequences on the computational algorithms used to analyse the digitized tissue samples, image processing techniques can be used to normalize the samples and get a more consistent dataset [8]. This is

most commonly referred within digital histopathology as color management, see the pre-processing step in Fig. 1.

5) Oral mucosal tissue structure: The oral mucosa consists of two main layers, the epithelial layer and the lamina propria [19]. The epithelial layer is the outer-most layer and is formed by epithelial cells (Epith.). The lamina propria consists of multiple layers, the papillary layer and the underlying reticular layer [20]. They both contain fibroblast cells that produce collagen fibers. In the lower layer the cells are more spread out with thicker regions of collagen [5], [20]. Fibroblast cells (Fibr.) are present throughout lamina propria. Endothelial cells (Endo.) are lining vascular channels throughout lamina propria [20].

Lymphocytes are immune cells that appear in inflamed areas and are therefore not very present in healthy tissue. Both Inflammatory cells (Infl.) and Lymphocyte (Lymph.) are present in unhealthy tisse in areas of acute and chronic inflammation [20]. A large aggregation of lymphocytes is a sign of an active disease.

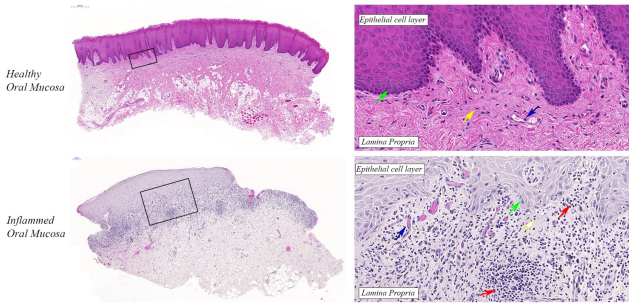

Fig. 2. Two WSIs of a healthy oral mucosa (top) and an inflamed oral mucosa (bottom). (Image provided by R. Sugars for Masters Thesis [5])

In Fig. 2 the black boxes locate the magnified areas on the WSIs. The arrows point to different cell types: Infl, (red), Epith. (green), Fibr. (yellow) and Endo. (blue).

Since some cells appear more frequently than others in the tissue, imbalance in the amount of labeled cells of different types is an inherent characteristic of histological datasets. As an example, Epith. cells appear much more in the tissue than Infl. cells since Epith. cells form the tissue structure while Infl. cells appear only in inflamed areas.

B. Color normalization

1) General: The term color normalization (CN) encompasses methods used to alter the color distribution of an image to fit certain needs. This can be achieved with many different methods such as histogram specification or generative adverserial networks [21], [22]. Within digital histopathology they can be applied as a pre-processing step to transform the tissue samples with variable colors to a common color space [2]. The aim of the CN algorithms in digital histopathology is to reduce the color variability in the dataset introduced by the staining. Since the color in digital histopathology images comes from staining, CN in digital histopathology is also commonly referred to as stain normalization.

2) Warwick toolbox: The CN methods in the Warwick toolbox [23] require a *target image* that defines the pursued color distribution for the methods to replicate on the input image.

The first type of CN method used is a color transfer method. In this method, the original RGB image is transformed to the perception-based $l\alpha\beta$ color space [21] and the mean and the variance is matched to the one from the target image. The method is implemented in the toolbox according to Reinhard et al. [21].

The second type of CN method uses a stain deconvolution method, introduced by Macenko et al. [15], it relies on a stain vector that represents the proportion of each wavelength absorbed and it characterises the stain present on the image. The challenge of stain deconvolution, however, is robustly estimating the stain vectors V, which should be done adaptively for each image [8]. This was further improved by Khan et al. [24] which is the state-of-the-art filter in this toolbox.

The Reinhard method was initially designed for general color transfer between images, while both Macenko and Khan are designed specifically for CN of digital histopathology images.

C. Machine learning

1) General: The use of Artificial Intelligence (AI), Machine learning (ML) and Deep learning (DL) has been called the fourth industrial revolution due to the fact that AI methods, tools and vocabulary are used in many fields to systematize and automate problems [25]. AI tools have become state-of-the-art in numerous medical applications by identification, quantification and classification of patterns in medical images to support practitioners [26], [27]. These developments make it possible to standardise and automate manual and subjective tasks, leading to more effective and efficient patient care [28].

2) Machine and deep learning models & the classification problem: The goal of ML models is to develop automated methods to detect and uncover patterns in data to make better predictions. This makes it similar to statistics, it differs primarily in its emphasis and terminology [29]. As Goodfellow et. al puts it ”Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions” [30]. This quote refers the *Universal approximation Theorem* presented in 1989 by Kurt Hornik [31] which states that a large enough ML model can estimate any complex function arbitrarily accurately. However, even though a ML model theoretically could approximate any function, that is far from the truth in practice. Usually this means that our ML-algorithm might not find the true value for our internal parameters, or it finds a wrong function [30].

Moreover, a DL model is a large ML model which deals with a vast number of parameters, in some cases even in the order of $10^6 - 10^7$ as for example with Google's EfficentNet [32]. This allows the algorithm to learn highly complicated patterns and requires a large amount of data [30].

ML methods can employ different types of learning: supervised, unsupervised and reinforcement learning. Supervised

learning tries to map inputs x to outputs y, given a training set $D = \{x_i, y_i\}_{i=1}^{N}$, where N is the number of data points. Each entry x_i is referred to as "features". This means estimating a function where a training set D of correctly identified observations is at your disposal [33]. Unsupervised learning however, tries to identify "interesting patterns" in data given no labels y_i, which make these problems less well-defined [29]. Supervised models address different types of classification problems, the most common are binary, multi-class and multi-labeled [34]. For multi-class problems, you have a training set D of different classes and our model predicts one of these classes for all data points in an unseen but similar dataset [35].

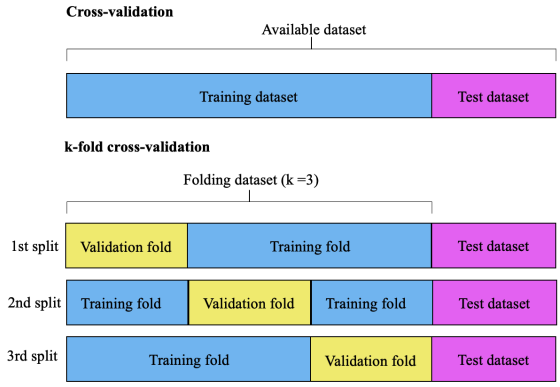

Fig. 3. A visualisation of the differences between test, validation and training set.

For ML models it is important to distinguish between training, validation and test data which is depicted in Fig. 3. Starting with a large data pool of available data, a portion of the data is taken out and marked as test data. The test data is used after training to understand how well the learned patterns generalize and is not used during training. The remaining data can then be split into training and validation. The training data is used to fit the internal parameters of the model, while the validation data is there to estimate generalization errors during training, optimize hyperparameters and to compare with the test data performance after training. This is called cross-validation. However, when a dataset is too small k-fold cross-validation is used instead, where the available dataset is partitioned in k splits randomly. If this is done by keeping the class distribution it is called stratified k-fold cross validation. By doing this we train k models to minimize the bias [30], [36].

All supervised ML algorithms contain essentially four properties: a model, internal (inside model) parameters and external parameters (also called hyperparameters), a notion of penalizing bad predictions per iterations (cost functions) and an optimization for minimization direction. Internal model parameters are specific for each model, for example a Neural Network model contains weight matrices W and biases b and a simple linear model has parameters slope k and m being y-intercept [30]. The hyperparameters are used to control the updating process of the internal parameters [30]. A cost (or loss) function is a real-valued function of the training and validation data. The cost function gives a numerical score where by convention a lower numerical score is better. Cost functions can be as exotic as cross-entropy loss or as simple as Mean Squared Error (MSE), known from undergraduate statistics class for linear regression models [37]. Something a statistician might call model fitting of internal parameters an ML researcher would call "learning" [38]. The training is, ideally, driving this numerical score towards gradually lower scores, so that the model learns. During the training process some model parameters grow out of scale, an activation function "restricts the values within an acceptable range" [39]. Its selection of activation function is dependent of the model and where in the model it is used. Common activation functions are Sigmoid, ReLU, ELU [39], [40]. Lastly, the cost function is minimized iteratively with an optimization algorithm. Since 1989 one of the main optimization algorithms has been gradient-based back-propagation [41], in short Back-prop. Backprop is essentially a numerical computation of the chain rule to compute the partial derivatives with respect to all internal parameters [30], [40].

Two common combinations of cost function and optimizer are Stochastic Gradient Descent (SGD) and Cross-Entropy Loss (CEL) mentioned above. CEL is the negative log-likelihood of our training labels, model parameters and input variable and has been found to lead to faster learning and improved generalization [30].

SGD is a common less computationally expensive solution in gradient-learning, to minimize a loss function L. When using SGD the gradient is estimated using a sample size n. One version of SGD uses two hyperparameters: learning rate l and momentum m. The pseudocode is presented in Algorithm 1. Momentum aims primarily to handle variances in the stochastic gradient due to our sampling method [30]. Learning rate is simply determining the size of the step. Most advanced learning algorithms adapt this parameter throughout the learning phase, for example by a decreasing scheme under the assumption that only incremental changes are needed later in the process when the minima is reached [30]. SGD is usually calculated on $32 - 515$ datapoints, since fewer datapoints tend to ease the learning process for DL models [42].

Algorithm 1: SGD with momentum [30]

Require: Learning rate l, momentum parameter m
Require: Initial parameter θ, inital velocity v
while *stopping criterion not met* **do**
  Sample a minibatch of m examples form the training set $x^{(1)}, ..., x^{(n)}$ with corresponding targets $y^{(i)}$
  Compute gradient estimate:
  $g \leftarrow \frac{1}{n} \nabla_\theta \sum_i L(f(x^{(i)}; \theta), y^{(i)})$
  Compute velocity update: $v \leftarrow mv - lg$
  Apply update: $\theta \leftarrow \theta + v$
end

3) Image classification models & Convolutional Neural Networks & EfficientNet: Image classification models form a subgroup of classification models that classify images. Among different approaches, Convolutional neural networks

(CNNs), a type of DL model, have become standard in image classification tasks and have achieved state-of-the-art results [43]. These neural networks use parameterized convolution kernels that preserve some of the spatial characteristics of the classified images [44].

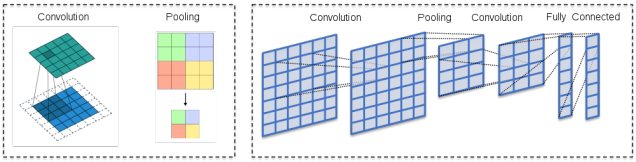

Fig. 4. Visual representation of convolution and pooling layers [45]

The convolution operation is done by sliding the kernel over the image and calculating the scalar product between the kernel and a specific part of the image. The kernel size defines the size of the image part to be scalar multiplied with the kernel. A convolutional layer of a CNN is made up of several of these kernels, where the conceptual idea for the different kernels is to learn local spacial features of the image [46]. To reduce the size of the input, pooling is then applied to the resulting convolutions. These two operations are visually represented in Fig. 4. A kernel in a CNN is a matrix randomly initialized which tries to abstract different features from the image. A feature is a specific representation of the image with some underlying pattern that seems to be evident by the algorithm. However, it is debatable what those features actually are representing for a human. Older versions of ML algorithms used to have hand-crafted features created by domain experts, for example edge detectors [1].

The number of parameters involved in these CNNs have since the success of AlexNet in 2012 [47] increased drastically to improve the accuracy of the models. The number of parameters is a compromise as too many parameters lead to a computationally expensive model and too few may compromise the results of the model. In the article *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* [32] Tan and Le introduce an effective compound scaling method that reaches state-of-the-art accuracy and can scale to any target constraints. Different scales of the EfficientNet model are introduced named B0 to B7. The architecture of the model with the least parameters, EfficientNet-B0 is presented in Tab. I where MBConv stands for mobile inverted residual bottleneck blocks, which have shown to make the process more efficient [32], [48].

A common practice in DL and specifically in image classification is to utilize already trained models on large community (open sourced) hand-labeled datasets, for example ImageNet [49]. Transfer learning means using some parts of a pre-trained model and then training it for a new but similar task [30].

4) Limited dataset & class imbalance & data augmentation & overfitting: The models and learning algorithms used today are nearly identical, at least conceptually, to those used 20-30 years ago. The difference is that the availability of large amounts of data have reduced the level of competence needed for the user. The performance of a DL model is highly dependent on the amount of data available during training [30]. When large amounts of data is not available the performance generalizability of the model becomes harder to adjust, especially if the dataset has an inherent class imbalance for representative examples. Models with low generalizability are overfitted and perform well only on the training data, which can be seen when comparing the validation and test accuracy since validation is a subset of the training set [44]. This means that when a model has not generalized well it has not learned broad enough features from the training data to be able to interpret variations in the test dataset. Models trained on imbalanced datasets show worse performance than those trained on balanced datasets especially for classification problems [50].

TABLE I
EFFICIENTNET-B0 ARCHITECTURE

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	224 × 224	31	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

One technique to handle class imbalance is to use oversampling, where we use multiple versions of the same data from the same class. By doing this we can balance out our dataset at a cost of increased risk for overfitting [50].

Moreover, there exists a lot of different label-preserving data augmentation techniques to minimize overfitting. A common method is random cropping, which involves resizing the sample and interpolating the new pixel values, to later randomly crop to a chosen size. Another method is noise injection, which adds imperceptible perturbations to the images. A specific noise is adversarial noise, this noise is specifically created to make the model make wrong predictions [51].

Regularization techniques minimize overfitting for limited datasets by using a weight decay that adds a regularization term to the cost function which supports the optimization algorithm getting closer to a minimum [30].

5) Ensemble learning: An ensemble learning method combines multiple model predictions into one prediction. The multiple models are trained independently and each of them are given a vote in evaluation of the test data. This has been found to increase generalization, minimize error in predictions and decrease variance of predictions [26], [30]. The most commonly used ensemble learning methods are majority voting, probability score averaging (PSA) and stacking ensemble [26].

6) Model evaluation methods: To evaluate empirical notions of accuracy and performance of a classification model different measures can be used to help assess the models ability to predict the classes of the unseen data.

To evaluate a classification model for a dataset with class imbalance it is standard to use Precision, Recall and F1-score

per n classes. To scale these measures to tell us something about the entire model accuracy, macro and weighted measures are used, see Eq. 1 -14 below. To understand these measurements it's more intuitive, and eases the notation, to start with the classification (confusion) matrix. A confusion matrix is a visualisation tool where the entries are basis for all accuracy metric calculations in classification problems arising in multiple fields from computer vision to natural language processing [35]. The matrix, see Tab. II, has entries $x_{i,j}$, column j being actual label and row i being model prediction, opposite conventions exists in some literature. Each entry is the number of datapoints with the predicted and actual label of that particular row and column.

TABLE II
GENERALIZED CLASSIFICATION (CONFUSION) MATRIX

	Class 1	Class 2	Class 3	...	Class N
Class 1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	...	$x_{1,N}$
Class 2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	...	$x_{i,N}$
Class 3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	...	$x_{i,N}$
⋮	⋮	⋮	⋮		⋮
Class N	$x_{N,1}$	$x_{N,2}$	$x_{N,3}$	...	$x_{N,N}$

For the binary classification problem our NxN dimension confusion matrix becomes a 2x2 matrix, see Tab. III. Each entry has an associated name, where $x_{1,1}$:= True Positive Count (TP), $x_{1,2}$:= False Positive Count (FP - Type 1 error), $x_{2,1}$:= False Negative Count (FN - Type 2 error) and $x_{2,2}$:= True Negative Count (TN) [7], [52].

TABLE III
BINARY CLASSIFICATION (CONFUSION) MATRIX

	Class 1	Class 2
Class 1	TP	FP
Class 2	FN	TN

We can now define our metrics for the multi-class and binary classification problem. For the binary classification problem, with only two classes the common measures are:

$$Accuracy := \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision := \frac{TP}{TP + FP} \tag{2}$$

$$Recall := \frac{TP}{TP + FN} \tag{3}$$

$$F_1 - score := 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

For each Eq 1 - 4 one can investigate each metrics purpose or "Evaluation focus", inspired by Sokolova (2009) [34]. This will give an intuition for each measure before we generalize to datasets with more classes than two. $Accuracy$ evaluates how well the model predicted the two classes, or the overall effectiveness of the model. $Precision$ is the fraction of the predicted positives which were actually positive. $Recall$ is the fraction of how many of the actual positives where predicted as such. When evaluating performance on an unbalanced datasets accuracy is a poor performance metric for characterizing and $Precision$ and $Recall$ give a better representation of model performance [30], [33]. In specific applications $Recall$ or $Precision$ could be especially important. For example, in digital pathology $Recall$ is important since false negatives could mean missing to diagnose a certain disease. Lastly, $F_1 - Score$ is a harmonic mean of $Precision$ and $Recall$ [52], which penalizes a spread amongst $Precision$ and $Recall$ in a better way than an arithmetic mean. For example, $Precision = 0.1$ and $Recall = 0.9$, would give $\overline{x} = 0.5$ while $F_1 - Score = 0.18$ which gives us a better representation of this poorly performing model.

For the multi-class classification problem there are N classes or states, which means that Eq. 1 - 4 has to be generalised for more than two classes. We will use the introduced notation from Tab. II.

$$Accuracy := \frac{\sum_{i=1}^{N} x_{i,i}}{\sum_{i=1}^{N} \sum_{j=1}^{N} x_{i,j}} \tag{5}$$

$$Precision_n := \frac{x_{n,n}}{\sum_{i=1}^{N} x_{n,i}} \tag{6}$$

$$Recall_n := \frac{x_{n,n}}{\sum_{i=1}^{N} x_{i,n}} \tag{7}$$

$$F_1 - Score_n := 2 \cdot \frac{Precision_n \cdot Recall_n}{Precision_n + Recall_n} \tag{8}$$

Note that measures above for $Precision$, $Recall$ and $F_1 - Score$ are per class n of N classes, denoted $Precision_n$, $Recall_n$ and $F_1 - Score_n$. A $Precision$, $Recall$ and $F_1 - Score$ measure for the entire model could also be of interest to give a notion of average $Precision$, $Recall$ and $F_1 - Score$. Most common approaches here is to combine each measurement across classes with an normal arithmetic average, also called macro average, here denoted ^{MA}, see Eq. 9-11.

$$Precision^{MA} = \frac{1}{N} \sum_{n=1}^{N} Precision_n \tag{9}$$

$$Recall^{MA} = \frac{1}{N} \sum_{n=1}^{N} Recall_n \tag{10}$$

$$F_1 - Score := 2 \cdot \frac{Precision^{MA} \cdot Recall^{MA}}{Precision^{MA} + Recall^{MA}} \tag{11}$$

Class imbalance can be accounted for by multiplying each measurement with its associate weight w_i (number of class appearance divided by total number of points), also called weighted macro average, here denoted ^{WMA} [34], see Eq. 12- 14.

$$Precision^{WMA} = \frac{1}{N} \sum_{n=1}^{N} w_n \cdot Precision_n \tag{12}$$

$$Recall^{WMA} = \frac{1}{N} \sum_{n=1}^{N} w_n \cdot Recall_n \tag{13}$$

$$F_1 - Score := 2 \cdot \frac{Precision^{WMA} \cdot Recall^{WMA}}{Precision^{WMA} + Recall^{WMA}} \quad (14)$$

7) Limits of DL algorithms and intrinsic variability: Much of the research in DL can be explained by the *no free lunch theorems* which state that each class of optimization problems need specific solutions [53]. Just because an DL algorithm works on a specific sub-problem and dataset it does not mean it will work on a similar setups, meaning there is no superior DL algorithm for all uses [30]. Another issue with DL models is variability. One of the most cited papers within this area, written by Dietterich [54], concluded in 1998 that there are multiple random sources causing the variability. These sources include random variation in data selection, internal randomness in common algorithms and random classification errors. This causes challenges for reproducibility within the academic community.

The training of the large models used in DL requires large computational resources and time allocation. Cutting edge research in the field often needs a high performance computer cluster for effective training [40].

Concerns have been raised about the unexplainable nature of decisions made by DL algorithms and a need for them to be interpretable by humans. This is called explainable AI and deals with methods that "enable causality, explanatory ability, and interpretive ability of the prediction results of the model" [55], [56], which may be important in some applications such as diagnostic tasks within the medical field.

D. Related work

In this paragraph relevant research will be presented as a basis for the discussion of this paper and a highlight of their findings. Pontalba et al. [8] used the CNN models named CNN3 [57] and UNET3 [58] for a segmentation task based on three different H&E stained datasets TCGA [57], TNBC [59] and SMH (not public) [8]. Stain normalization was used as a pre-processing step with the Warwick toolbox [23]. The papers showed that the filters introduced variability and used the ensemble method PSA. The paper only presented validation metrics with dice similarity coefficient (DSC) [8] as the main performance metric. The performance varied for the TCGA and TNBC dataset where the ensemble learner performed better than the individual filters with the DSC metric.

Estreen [7] used the EfficientNet model [32] on the H&E-stained KI dataset. However, Estreen did not use the same training/validation/test composition as this report. The report also displayed validation metrics of accuracy $\sim 92\%$ as the main result. Within the same research group Brynjarsson [5] used shallow neural networks on the KI dataset, VGG16, RCCNet & Softmax. The data composition was slightly different to this report. The main presented results were in terms of validation accuracy where the architectures performed as following: VGG16: $\sim 85\%$, RCCNet: $\sim 88\%$ & Softmax: $\sim 90\%$.

Following will be an exposition of previously reported results from object detection models from similar fields. Chouhan et al. [60] showed an increase of ~ 2 percentage points on accuracy from their best performing model on a similar task using a majority voting technique on a classification task. Shorfuzzman et al. [61], similarly to Chouhan et al. showed an increase of ~ 1 percentage point. Lakhani et al. [62] showed an increase of ~ 1 percentage point on AUC (Area under the ROC Curve) from their best performing detection model with PSA. Hooda et al [63] showed a ~ 4 percentage points increase on a similiar task as Lukani et al. presented. Hinton et al. [64] reported interesting results on a speech recognition model using ensemble learners. The use of ten models for ensembles only increased the accuracy ~ 2 percentage points.

Islam et al. [65] showed that ensemble learning models experience a diminishing return on investment at around 5-10 models. We cannot continue to add models and expect the performance to continuously increase. However, the robustness of the model improves as more models are added. PSA is also expected to perform better than majority voting.

III. Methods

The process pipeline consisted of four different parts that we are accounted for in order here. The process setup is visualized in Fig. 5 and each component is addressed below.

A. KI Dataset

The dataset consists of partially labeled 2000 x 2000 pixel WSI of tissue samples from six patients. The tissues have a great variety of histological grade (G0-G4) [14] and quality of the H&E staining.

The images were labeled by pathologists at KI with an open source tool called LabelIMG [66]. The tool generates markup-files (.xml) with each labeled object having a name (cell type) and location (bounding box). Four different types of cells were labeled: Inflammatory (Infl.), Lymphocyte (Lymph.), Fibroblast and Endothelial (Fibr./Endo.) and Epithelial (Epith.). See Tab. IV-VII for their specific occurrences in the dataset, Fig. 6 for a visualization of their location in the WSI and Fig. 7 for a visualization of the individual cropped cell images.

TABLE IV
Training set

Grading	Patient ID	Images	Image labels
G0	P20	21	13 121
G3	P9	7	1 261
G4	P19	6	1 234
Tot.		34	15 616

TABLE V
Training set per cell type

Grading	Patient ID	Infl.	Lymph.	Fibr./Endo.	Epith.
G0	P20	420	935	3 745	8 021
G3	P9	148	404	284	425
G4	P19	147	628	199	260
Tot.		715	1967	4 228	8 706

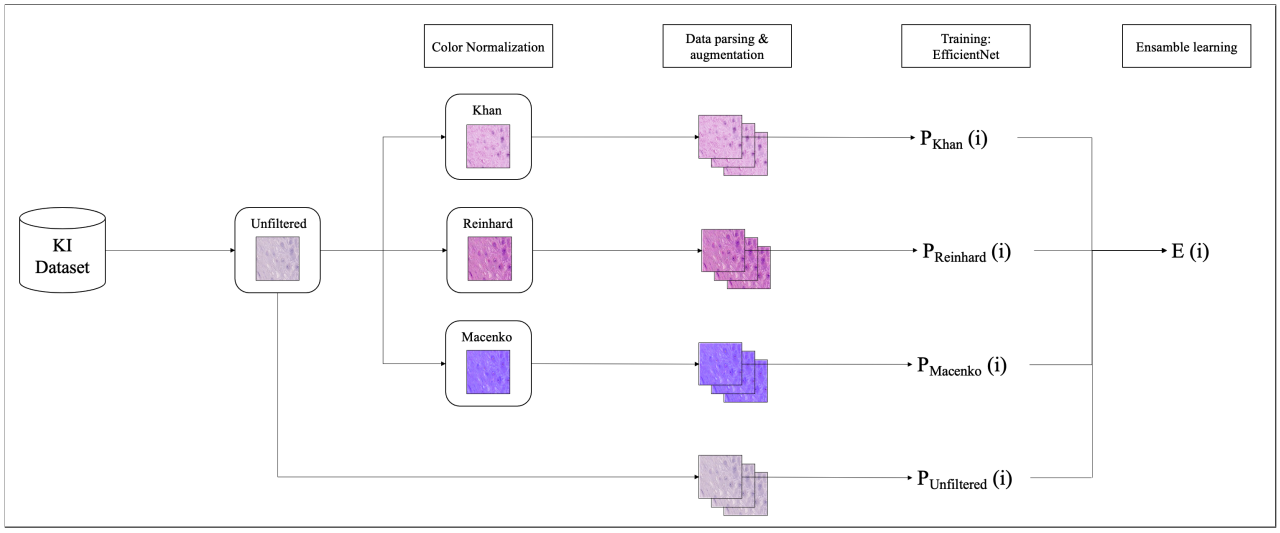

Fig. 5. Visualization of project pipeline

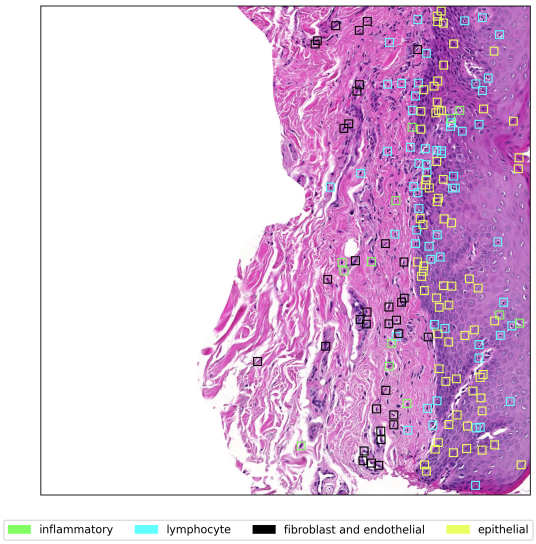

Fig. 6. Visualization of location on type of the annotated cells in the 2000x2000 slide P9_4_1

TABLE VI
TESTING SET

Grading	Patient ID	Images	Image labels
G0	N10	5	2 094
G3	P13	4	2 004
G4	P28	4	7 220
Tot.		13	11 318

B. Color Normalization

All images were filtered using three publicly available CN algorithms based on *Stain Normalization Toolbox* provided by the Department of Computer Science of the University of Warwick [23]. These three filters are referred to as Macenko [15], Reinhard [21] and Khan [24]. The image P9_4_1 from the KI Dataset was used as a reference image for all three CN algorithms. This specific image was chosen because it showed good staining quality. Every single image in the dataset was then filtered with the three filters and saved as four different

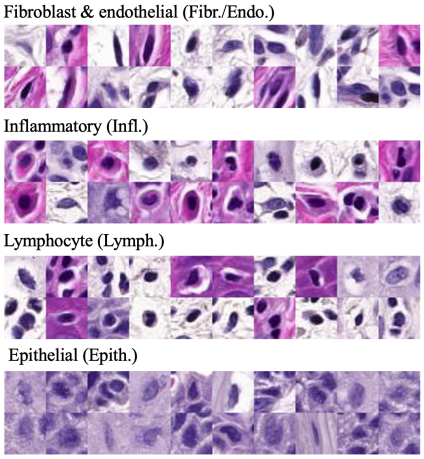

Fig. 7. Visualization of different cell types in our dataset. The cells are cropped from images P20_5_1, P9_3_1 and P19_2_1.

TABLE VII
TESTING SET PER CELL TYPE

Grading	Patient ID	Infl.	Lymph.	Fibr./Endo.	Epith.
G0	N10	79	59	634	1 322
G3	P13	333	278	787	606
G4	P28	729	1 987	1 775	2 729
Tot.		1 141	2 324	3 196	4 657

image files (one unfiltered, and one for each of the three filters).

C. Data parsing & augmentation

The 2000x2000 WSIs are cropped to 32x32 images around the center of the labeled cells, creating a separate image per label (see Tab. IV & VI and Fig. 7). Note that unlabeled cells were not taken into account.

All parsed images were Gaussian normalized to minimize the impact of intensity and contrast variability, and transformed pixel means above 0.95 were considered white and ignored [7].

A limited dataset is prone to issues with overfitting, less generalizability and biased models. To address these first two problems, five data augmentation methods were used on the training and validation set. The first method is random cropping and involved resizing the images from 32x32 to 246x246, the new pixels are interpolated using bicubic interpolation [67], [68]. After the resizing to 246x246, the images are cropped around a randomly chosen location to 224x224. In the second method, Gaussian Noise was added to all pixel values ($\mu = 0 \,\&\, \sigma = 0.1$). Thirdly, additional training samples were created by injecting adversarial noise (AdvProp). Fourthly, a regularization technique was used together with the chosen optimizer, which set weight decay to 0.00005. Lastly, also due to the selection of small-batch regime optimizer, the batch size was set to 32. To address the bias associated with a limited dataset, stratified k-fold cross validation techniques were used with $k = 5$. Because of the imbalanced dataset, oversampling techniques were used to equalise the magnitudes over our four class (cell) types. This oversampling involved augmenting the dataset, see Tab. VIII.

TABLE VIII
TRAINING SET PER CELL TYPE - BEFORE AND AFTER OVERSAMPLE

Total cell type count	Infl.	Lymph.	Fibr./Endo.	Epith.
Before oversample	715	1 967	4 228	8 706
After oversample	6 435	9 835	8 495	10 706
Scaling factor	900%	500%	200%	123%

D. EfficientNet: Cell classification model

The cell classification model is based on Google's *State-of-The-Art* Convolution Neural Network EfficientNet [32] through an early January 2020 open source implementation [69], using the model architecture with least parameters: EfficientNet-B0. The implementation is using Facebook's open source deep learning framework called PyTorch [70]. Weights were updated under training and cross entropy loss was selected as the cost function. To ease the computational time Stochastic Gradient Descent (SGD) was used to estimate the actual minimization direction. The learning rate, the cost function multiplier, was set to 1%. A learning rate schedule was used by decaying it 3% every 2.4 epochs (iterations), to converge to the minimum in a smoother manner. To accelerate the training and to get better performance with fewer epochs, momentum was set to 0.9. Transfer learning was used by loading a model pretrained on the ImageNet dataset [7].

E. Training and implementation

For each of the 5 validation splits 100 iterations were used and the computation time for each trained model with selected hyperparameters was in the magnitude of 8 hours. Over the 100 epochs the model with the highest validation Top1-accuracy was chosen. Computations were made using NVIDIA V100 and T4 nodes on a Swedish National Infrastructure for Computing (SNIC) resource named Alvis [71] through the grant agreement no. 2020/33-67. The entire project is documented and available at github [72].

F. Ensemble learning

Two ensemble learning regimes were implemented based on all five trained models (k=5, see above) for each four datasets with six iterations, giving us a total of 120 models to combine. The first algorithm was based on a naive approach where each models top1-class prediction per image was combined via a majority voting technique. This means that if three models predicted Lymph. and one Infl., Lymph. was predicted. We dealt with equal predictions by uniformly randomizing the prediction, see Algorithm 2.

In the following pseudocode explanation, N refers to the number of different filters used, including the unfiltered version, in our case $N = 4$. M refers to number of data points in the dataset, in our case the testing set has $M = 11\,318$.

Algorithm 2: Naive majority voting ensembling

Input : N vectors y_j where $y_j(i)$ is the Top1 predicted class of the i:th data point
Output: the vector e where $e(i)$ is the ensemble predicted class of the i:th data point
for $i \leftarrow 1$ **to** M **do**
  Assign to $e(i)$ the most common value $y_j(i)$ for j between 1 and N;
  If two or four values appear the same amount of times, we randomly choose one of the two or four
end

The second more restrictive and representative regime PSA was inspired by the Pontalba paper [8], [26], see Algorithm 2. A softmax activation function was applied to the output layer, giving us four probability vectors (P_i) for each image and model. These were added up, and scaled down by model count, see Eq. 15.

$$E_i = \frac{1}{N} \sum_{filter} P_i \tag{15}$$

G. Model evaluation

To evaluate the model precision, recall and F1-score were used on the top1 prediction for each cell type, including an average for all cell types, with and without weights. Accuracy refers to Top-1 accuracy which uses the Top-1 class, the class with the highest predicted probability. The Top-1 class is also used to calculate precision and recall. Confusion matrices were added to strengthen the analysis per class. Lastly, loss and

Algorithm 3: PSA ensembling

Input : N matrices p_j where $p_j(i)$ is the vector with the predicted class probabilities of the i:th data point

Output: the vector e where $e(i)$ is the ensemble predicted class of the i:th data point

for $i \leftarrow 1$ **to** M **do**

  Assign to V the sum of $p_j(i)$ for j between 1 and N;

  Scale down the V with N for each element.

  Assign to $e(i)$ the top1 predicted class of V

end

accuracy curves were used to study the learning process during the training phase.

IV. Results

A. Color Normalization

The first step where results could be seen was during the pre-processing step. Depicted in Fig. 8 we show some representative parts of the color normalized dataset over a variety of histological severity and oral mucosa structures.

B. Classification Performance Metrics

The 120 trained models were all evaluated on the testing set filtered in the same manner as the training set. The selection of models for the ensemble method were arbitrary as long as they were unique and one from each filter. No model was used in the ensemble learning more than once. Presented are our findings with our independent variable being the filters and all other parameters were fixed during our experiments. The evaluation metrics from these classifications are presented in the Fig. 9 - 12.

In Fig. 9, the boxplots show the accuracy per filtered model type for the four differently filtered datasets and the two ensemble learning techniques. The second Fig. 10 gives additional insights with the metrics precision, recall and F1-score which was averaged in two different ways. Fig. 11 shows how all model types performed over our four cell types. In Fig. 12 the relationship per model type is presented. This figure aims to investigate if a certain filter helps the trained model to classify a certain type of cell.

To further visualize the results, the metrics of the best performing model in each model type is presented in Appendix A. The appendix includes a confusion matrix to provide a better understanding of the classification. To show the learning process over epochs the appendix also includes loss and accuracy curves.

V. Discussion

A. Color normalization

The quality of the CN results differed between filters but also between different files when the same filter was used, see Fig. 8. It is possible that for each filter, artifacts from CN could arise. Both the filters Khan and Reinhard seem to produce a result that mimics a high quality H&E staining. Both the above statements are similar to the results presented by Pontalba et al. [8]. However, the filter Macenko introduced artifacts for some image files that manifested as a very blue color that did not relate to the original image in any apparent way and as yellow and red patches on some parts of the image. Pontalba et al. [8] also referred to Macenko as showing "inconsistent color mapping". However the filter was kept to add variety in our results and to study its impact on model performance. According to Khan et al. [24] their filter is state of the art due mainly to three reasons: less introduction of artifacts, robustness and appropriateness for H&E staining. Khan et al. also claim that color deconvolution based methods, such as Macenko and Khan, are most appropriate for stain normalization since the "chemical processes are largely independent for each stain, and color deconvolution separates out the effect of variation of each stain so it can be corrected independently". Khan et al. [24] propose two reasons why Khan outperforms Macenko: "1) Better, or more robust, deconvolution matrix estimation, and/or 2) a more appropriate mapping function." Furthermore, Khan et al. [24] argue that the Reinhard filter is "attractive in its simplicity but is based on the false assumption of unimodal color distribution in each channel" which is not true for dyes and stains.

Images that mimic high quality staining are expected to lead to better model training and classification. Conversely, inaccurate coloring and introduced artifacts are expected to negatively affect model training and cell classification.

B. Classification Performance Metrics

All filtered model types show an increased spread in accuracy values in relation to the non-filtered baseline model, see Fig. 9.

Even though Khan and Reinhard showed similar visual staining quality, the models trained on them performed vastly differently. Khan showed the highest accuracy and averaged metrics out of any of the individually filtered model types, see Fig. 9 and Fig. 10. On the other hand, Reinhard performed similarly to the unfiltered model type in accuracy and averaged metrics. The models trained on the Macenko filtered dataset had lower accuracy and averaged metrics than all other model types, see Fig. 9. The poor performance can perhaps be explained by the introduced blue artifacts of the Macenko filter. The artifacts might be due to the assumption of unimodal distribution presented in Ch. V-A but it needs further investigation for a definite conclusion to be drawn.

Both ensemble learners showed more robust accuracy values than any of the individually filtered models, see Fig. 9. The PSA ensemble learners outperformed the naive approach for accuracy and for the averaged metrics in Fig. 10. This was expected since PSA utilizes more information from each model. These results are consistent to what was presented in Ch. II-D. The ensemblers with 30 included models also showed very similar accuracy values to the best performing filter, Khan. This is interesting since the ensemble learners take into account the results from the poorly performing Macenko filter. This would indicate that further improvements to the

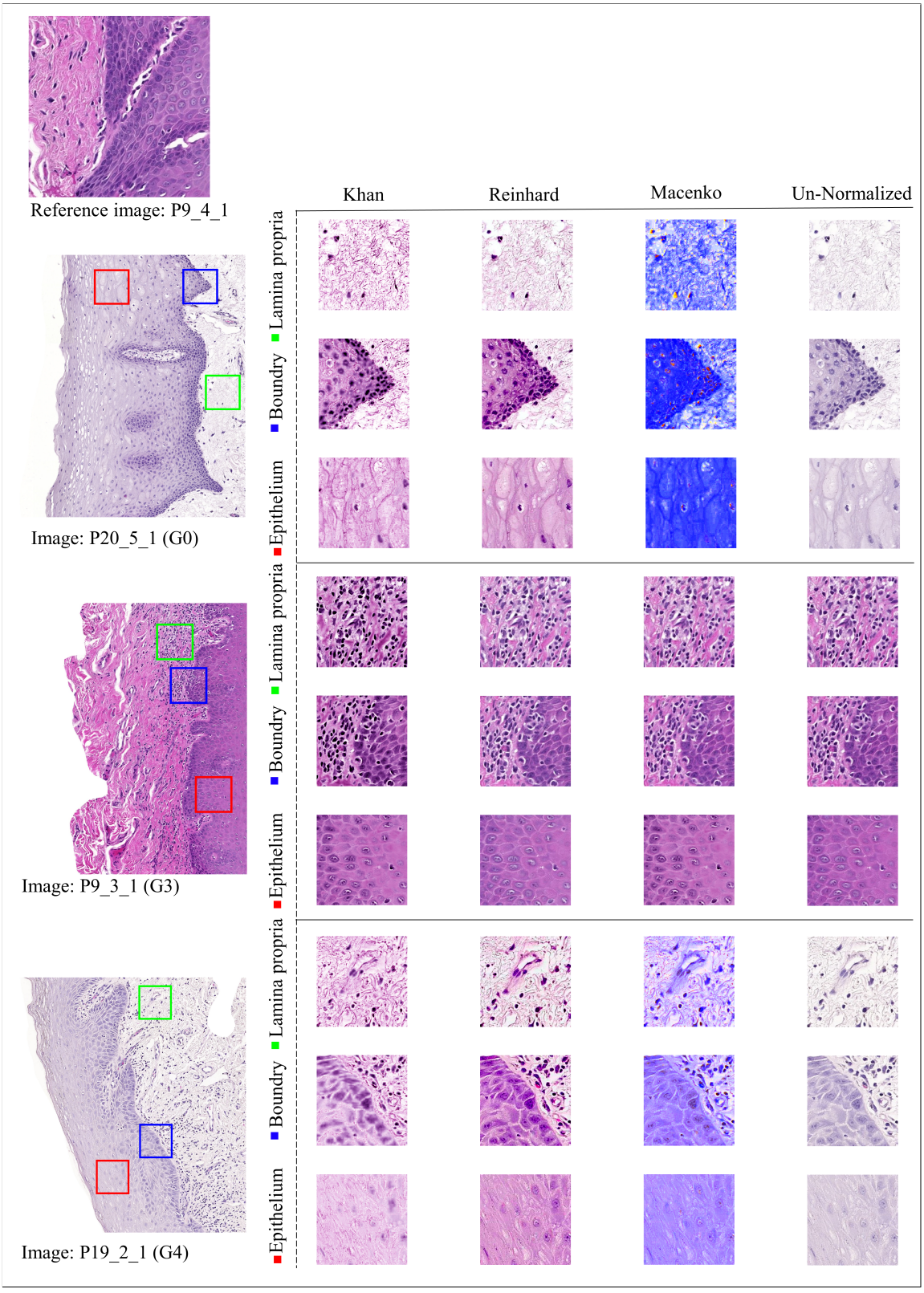

Fig. 8. CN filter results with varieties in histological grading (G0, G3 and G4) and oral mucusa structure: lamina propria (green), epithelium (Red) and the boundary (blue) in-between them. Meaning, for each image in the column "Un-Normalized" the three filtered results (Khan, Reinhard and Macenko) are depicted to the left with respect to a representative reference image.

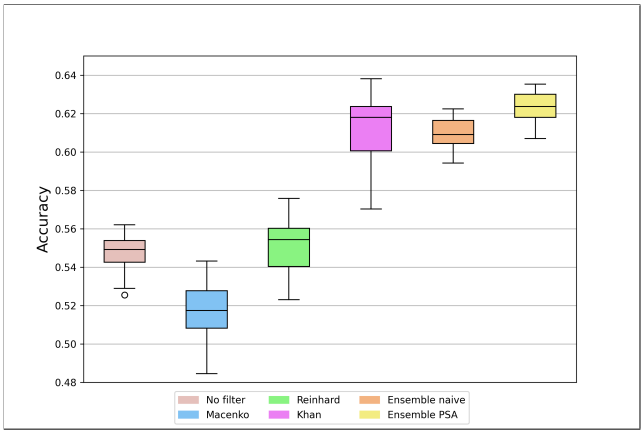

Fig. 9. Boxplot of accuracy results per filter and ensemble learning method. Evaluated on the test set, each box contains 30 data points (bincount).

performance could be made if models for the ensemblers are hand-picked and the Macenko filter is left out.

There is a clear trend for higher quality classification for cell types with more labels in the training dataset, see Tab VIII and Fig. 12. Specifically, for the inflammatory cell type all filtered datasets including non-filtered showed low quality classifications, see Fig. 11 and Fig. 12. This means that the used data augmentation techniques were not sufficient to prevent low quality classification for the rarest cell types.

One outlier from the general trend in filter performance was the classification of Lymphocytes where precision as expected was higher for Khan, but the recall was highest for Macenko. For classification of Epithelial cells the precision scores for Khan outperformed both ensemble learning techniques. Another trend shift can be seen for the inflammatory cell types where the ensemblers got worse metrics than the individual filters for recall, leaving the unfiltered dataset with the highest F1-score, see Fig. 12.

It is also clear that our model is highly overfitted when comparing the metrics from the validation and test set in Appendix A, where a 40 percentage points difference in accuracy values between validation and test datasets can be observed. The authors find it difficult to compare their results directly to Brynjarsson [5] who presents the validation metrics as the main result. However, the validation results in this report are in the same magnitude as both Estreen and Brynjarsson with a validation accuracy of $\sim 90\%$. The test accuracy presented by Estreen [7] is also similar to the test accuracy of this report, around $\sim 65\%$.

VI. Conclusions

For this project, three CN algorithms were used as a preprocessing step on the KI dataset to improve an existing EfficentNet CNN cell classification model. The CN algorithms were used to handle the color variability in the KI dataset. Performance was assessed by analysing the results from the individually trained models and by combining these results with ensemble learning techniques. Our conclusions are clear, stain normalization filters significantly impact classification performance. When we have a filter that introduces artifacts, such as the Macenko filter, the classification performance is worse than that of the unfiltered baseline. For filters with adequate staining qualities such as Khan, the performance is enhanced. Lastly, ensemble learning techniques have been shown to average out badly performing filters and giving us a robust performance, comparable to the best filter. We can conclude that the combination of well designed and selected CN methods and ensemble learning techniques boost performance for a cell classification model.

VII. Future work

Firstly, there are some issues in the existing pipeline that needs to be addressed. The oversample scheme is currently flawed and creates misleading validation metrics. The oversample process is performed on the training set before the training validation split. Therefore, some images can appear in both the training and the validation set multiple times. The chosen hyperparameters of the model need to aim for good performance on the test set rather than the validation set. Another main issue with the setup is our dataset that is still very limited making the class imbalance an especially difficult problem. There are shared and publicly available histopathological imaging datasets that are used in research [73] that could be used, however, none of these cover oral tissue samples specifically. It would therefore be of interest to expand the current dataset with more hand-labeled data possibly by collaborating with other institutes.

Ethics statement

The color normalization algorithms used in this paper are open source, and the KI dataset has been anonymized and does therefore not contain any information that could be linked back to any individual. Approval for the use of the image set of oral mucosal digitized histological slides for machine learning was granted by the Swedish Ethical Review Authority (Etikprövningsmyndigheten) Dnr: 2019-01259.

Appendix A
Scorecards per model

Acknowledgment

The computations and data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE Chalmers University of Technology partially funded by the Swedish Research Council through grant agreement no. 2020/33-67. Research funding was also provided by ALF Medicine and SOF Clinical Odontological Research Funding.

The authors would like to thank Karl Meinke and Rachael Sugars with team for their continued support and guidance during this project. The authors would also like to mention and thank Miritt Zisser at KTH Library for her support with the reference manager system Mendeley and our research process. Finally the authors thank their respective families for their continued encouragement and support during the process of this thesis. This accomplishment would not have been possible without the help of any of the aforementioned people.

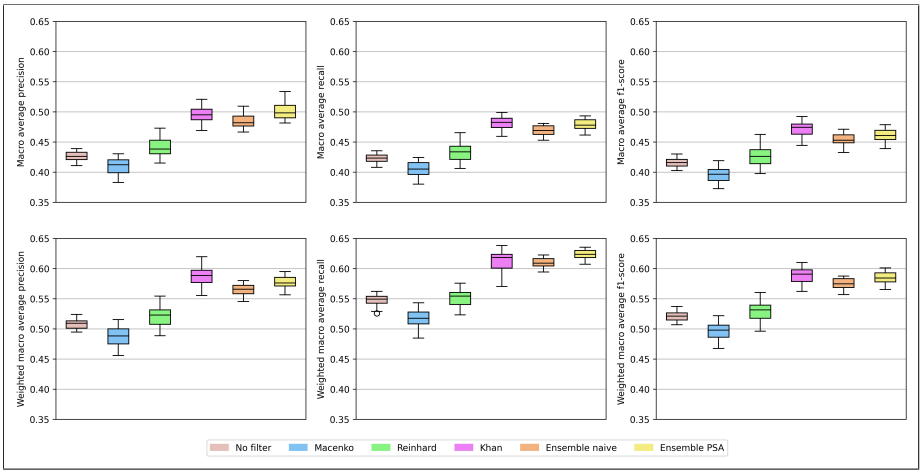

Fig. 10. Boxplot of combined (over all 4 cell types) precision, recall and F1-score values using macro average and weighted macro average. Evaluated on the test set, each box contains 30 data points (bincount).

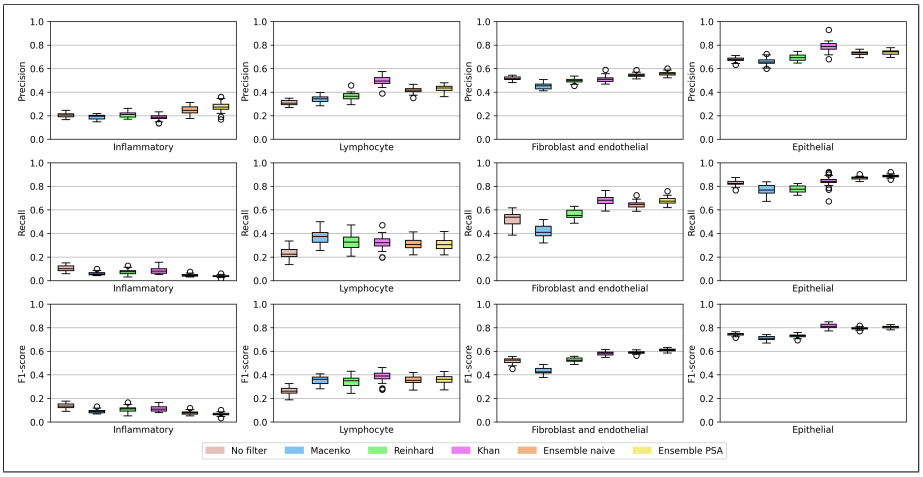

Fig. 11. Boxplot of precision, recall and F1-score per cell type. Evaluated on the test set, each box contains 30 data points (bincount).

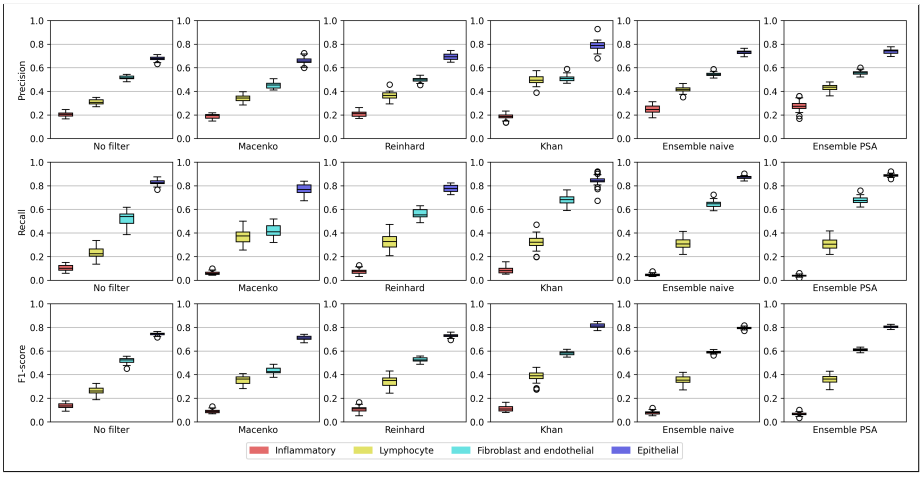

Fig. 12. Boxplot of precision, recall and f1-score per filter and ensemble learning method. Evaluated on the test set, each box contains 30 data points (bincount).

REFERENCES

[1] S. Deng, X. Zhang, W. Yan, E. I. Chang, Y. Fan, M. Lai, and Y. Xu, "Deep learning in digital pathology image analysis: a survey," *Frontiers of Medicine*, vol. 14, no. 4, pp. 470–487, Jul. 2020.

[2] B. Smith, M. Hermsen, E. Lesser, D. Ravichandar, and W. Kremers, "Developing image analysis pipelines of whole-slide images: Pre- and post-processing," *Journal of Clinical and Translational Science*, vol. 5, no. 1, pp. 1–11, Aug. 2021.

[3] S. Sayed, W. Cherniak, M. Lawler, S. Tan, W. Sadr, N. Wolf, S. Silkensen, N. Brand, L.-M. Looi, S. Pai, M. Wilson, D. Milner, J. Flanigan, and K. Fleming, "Improving pathology and laboratory medicine in low-income and middle-income countries: roadmap to solutions," *The Lancet*, vol. 391, Mar. 2018.

[4] M. Wilson, K. Fleming, M. Kuti, L.-M. Looi, N. Lago, and K. Ru, "Access to pathology and laboratory medicine services: a crucial gap," *The Lancet*, vol. 391, no. 10133, pp. 1927–1938, Mar. 2018.

[5] G. R. Brynjarsson, "Classifying nuclei in soft oral tissue slides," Master's thesis, KTH, Stockholm, Sweden, 2019.

[6] (2021, Apr.) Kebnekaise, swedish national infrastructure for computing. [Online]. Available: https://www.snic.se/resources/compute-resources/kebnekaise/

[7] T. Estreen, "Epithelial Layer Boundary Detection Using Graph Convolutional Networks for Digital Pathology," Master's thesis, [unpublished], KTH, Stockholm, Sweden, 2020.

[8] J. T. Pontalba, T. Gwynne-Timothy, E. David, K. Jakate, D. Androutsos, and A. Khademi, "Assessing the impact of color normalization in convolutional neural network-based nuclei segmentation frameworks," *Frontiers in Bioengineering and Biotechnology*, vol. 7, p. 300, 2019.

[9] A. Khademi, "Image analysis solutions for automatic scoring and grading of digital pathology images," *Canadian Journal of Pathology*, vol. 5, no. 2, pp. 51–55, Jun. 2013.

[10] N. Farahani, A. V. Parwani, and L. Pantanowitz, "Whole slide imaging in pathology: advantages, limitations, and emerging perspectives," *Pathology and Laboratory Medicine International*, vol. 7, pp. 23–33, Jun. 2015.

[11] J. Griffin and D. Treanor, "Digital pathology in clinical use: Where are we now and what is holding us back?" *Histopathology*, vol. 70, pp. 134–145, Jan. 2017.

[12] T. A. Azevedo Tosta, P. R. de Faria, L. A. Neves, and M. Z. do Nascimento, "Computational normalization of H&E-stained histological images: Progress, challenges and future potential," *Artificial Intelligence in Medicine*, vol. 95, pp. 118–132, Apr. 2019.

[13] H. Tizhoosh and L. Pantanowitz, "Artificial intelligence and digital pathology: Challenges and opportunities," *Journal of Pathology Informatics*, vol. 9, Nov. 2018.

[14] V. Tollemar, N. Tudzarovski, G. Warfvinge, N. Yarom, M. Remberger, R. Heymann, K. Garming Legert, and R. V. Sugars, "Histopathological Grading of Oral Mucosal Chronic Graft-versus-Host Disease: Large Cohort Analysis," *Biology of Blood and Marrow Transplantation*, vol. 26, no. 10, pp. 1971–1979, Oct. 2020.

[15] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas, "A method for normalizing histology slides for quantitative analysis," *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1107–1110, Jun. 2009.

[16] Wikipedia contributors. (2021, Apr.) H&E stain. Wikipedia. [Online]. Available: https://en.wikipedia.org/w/index.php?title=H%26E_stain&oldid=1017091635

[17] (2021, Apr.) What is H&E? University of Leeds Faculty of Biological Sciences, Leeds, United Kingdom. [Online]. Available: https://histology.leeds.ac.uk/what-is-histology/H_and_E.php

[18] M. R. Wick, "The hematoxylin and eosin stain in anatomic pathology—An often-neglected focus of quality assurance in the laboratory," *Seminars in Diagnostic Pathology*, vol. 36, no. 5, pp. 303–311, Sep. 2019.

[19] A. Nanci, "Chapter 1 - structure of the oral tissues," in *Ten Cate's Oral Histology*, 8th ed. St. Louis, Missouri: Mosby, 2013, pp. 1–13.

[20] ——, "Chapter 12 - oral mucosa," in *Ten Cate's Oral Histology*, 8th ed. St. Louis, Missouri: Mosby, 2013, pp. 278–310.

[21] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *Proceedings of the IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, Oct. 2001.

[22] M. T. Shaban, C. Baur, N. Navab, and S. Albarqouni, "Staingan: Stain style transfer for digital histological images," *IEEE 16th International Symposium on Biomedical Imaging*, pp. 953–956, Apr. 2019.

[23] D. Magee. (2021, Apr.) Stain normalization toolbox. Department of Computer Science at the University of Warwick, Warwick, United Kingdom. [Online]. Available: https://warwick.ac.uk/fac/cross_fac/tia/software/sntoolbox

[24] A. M. Khan, N. Rajpoot, D. Treanor, and D. Magee, "A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1729–1738, Jan. 2014.

[25] S. J. Russell and P. Norvig, *Artificial Intelligence A Modern Approach*. Englewood Cliff, New Jersey: Prentice Hall, 1995.

[26] S. T. H. Kieu, A. Bade, M. H. A. Hijazi, and H. Kolivand, "A Survey of Deep Learning for Lung Disease Detection on Medical Images: State-of-the-Art, Taxonomy, Issues and Future Directions," *Journal of Imaging*, vol. 6, no. 12, Dec. 2020.

[27] A. S. Sultan, M. A. Elgharib, T. Tavares, M. Jessri, and J. R. Basile, "The use of artificial intelligence, machine learning and deep learning in oncologic histopathology," *Journal of Oral Pathology and Medicine*, vol. 49, no. 9, pp. 849–856, May 2020.

[28] M. K. K. Niazi, A. V. Parwani, and M. N. Gurcan, "Digital pathology and artificial intelligence," *The Lancet Oncology*, vol. 20, no. 5, pp. e253–e261, May 2019.

[29] M. Borovcnik, H.-J. Bentz, and R. Kapadia, "A Probabilistic Perspective," in *Chance Encounters: Probability in Education*. Dordrecht, Netherlands: Springer-Verlag, 1991, pp. 27–71.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: MIT Press, 2016.

[31] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Mar. 1989.

[32] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *International Conference on Machine Learning*, pp. 6105–6114, Jun. 2019.

[33] J. Miao and W. Zhu, "Precision-recall curve (PRC) classification trees," *Evolutionary Intelligence*, pp. 1–25, Apr. 2021.

[34] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, Jul. 2009.

[35] A. Hinterreiter, P. Ruch, H. Stitz, M. Ennemoser, J. Bernard, H. Strobelt, and M. Streit, "ConfusionFlow: A model-agnostic visualization for temporal analysis of classifier confusion," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, Jul. 2020.

[36] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808*, Nov. 2018.

[37] E. Stevens, L. Antiga, and T. Viehmann, "Chapter 1," in *Deep learning with PyTorch*, 1st ed. Shelter Island, New York: Manning Publications Company, 2020.

[38] R. S. Michalski J. G. Carbonell T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*. Berlin, Germany: Springer Verlag, 1983.

[39] E. C. Too, L. Yujian, P. K. Gadosey, S. Njuki, and F. Essaf, "Performance analysis of nonlinear activation function in convolution neural network for image classification," *International Journal of Computational Science and Engineering*, vol. 21, no. 4, pp. 522–535, Apr. 2020.

[40] Y. Lecun, "1.1 Deep Learning Hardware: Past, Present, and Future," *IEEE International Solid-State Circuits Conference*, pp. 12–19, Feb. 2019.

[41] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[42] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," *ICLR 2017*, pp. 1–16, Feb. 2017.

[43] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "Breast cancer histopathological image classification using Convolutional Neural Networks," *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2560–2567, Jul. 2016.

[44] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, Jul. 2019.

[45] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Zeitschrift fur Medizinische Physik*, vol. 29, no. 2, pp. 86–101, May 2019.

[46] K. O'shea and R. Nash, "An Introduction to Convolutional Neural Networks," *arXiv preprint arXiv:1511.08458v*, Nov. 2015.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, Jan. 2012.

[48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Jan. 2018.

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

[50] P. Hensman and D. Masko, *The Impact of Imbalanced Training Data for Convolutional Neural Networks*, Bachelor's thesis, KTH, Stockholm, Sweden, May 2015.

[51] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 816–825, Jun 2020.

[52] D. Olson and D. Delen, *Advanced Data Mining Techniques*. Heidelberg, Germany: Springer-Verlag, 2008.

[53] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," *Natural Computing Series*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[54] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.

[55] M. Lee, J. Jeon, and H. Lee, "Explainable AI for domain experts: a post Hoc analysis of deep learning for defect classification of TFT–LCD panels," *Journal of Intelligent Manufacturing*, Mar. 2021.

[56] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[57] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, "A dataset and a technique for generalized nuclear segmentation for computational pathology," *IEEE Transactions on Medical Imaging*, vol. 36, no. 7, pp. 1550–1560, Mar. 2017.

[58] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Oct. 2015.

[59] P. Naylor, M. Laé, F. Reyal, and T. Walter, "Nuclei segmentation in histopathology images using deep neural networks," *IEEE 14th International Symposium on Biomedical Imaging*, pp. 933–936, Apr. 2017.

[60] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. de Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest X-ray images," *Applied Sciences*, vol. 10, no. 2, Jan. 2020.

[61] M. Shorfuzzaman and M. Masud, "On the detection of covid-19 from chest x-ray images using cnn-based transfer learning," *Computers, Materials and Continua*, vol. 64, no. 3, pp. 1359–1381, Jun. 2020.

[62] P. Lakhani and B. Sundaram, "THORACIC IMAGING: Deep Learning at Chest Radiography Lakhani and Sundaram," *Radiology*, vol. 284, no. 2, pp. 574–582, Aug. 2017.

[63] R. Hooda, A. Mittal, and S. Sofat, "Automated TB classification using ensemble of deep architectures," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31 515–31 532, Nov. 2019.

[64] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, Mar. 2015.

[65] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf, "Abnormality detection and localization in chest x-rays using deep convolutional neural networks," *arXiv preprint arXiv:1705.09850*, May 2017.

[66] T. Lin. (2018, Apr.) LabelIMG. GitHub. [Online]. Available: https://github.com/tzutalin/labelImg

[67] Wikipedia contributors. (2021, May) Bicubic interpolation. Wikipedia. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bicubic_interpolation&oldid=1005441439

[68] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.

[69] L. Melas-Kyriazi. (2020, Apr.) Efficientnet-pytorch. GitHub. [Online]. Available: https://github.com/lukemelas/EfficientNet-PyTorch

[70] (2021, Apr.) PyTorch. Facebook. [Online]. Available: https://ai.facebook.com/tools/pytorch

[71] (2021, Apr.) Alvis, Swedish National Infrastructure for Computing (SNIC). Chalmers University of Technology, Gothenburg, Sweden. [Online]. Available: https://www.snic.se/resources/compute-resources/alvis/

[72] A. Aillet and F. Frisk. (2021) Bachelor thesis documentation: Assessing the impact of stain normalization on a cell classification model in digital histopathology. Github. [Online]. Available: https://github.com/filipfusk/BscThesis

[73] D. Komura and S. Ishikawa, "Machine Learning Methods for Histopathological Image Analysis," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 34–42, Jan. 2018.