RESEARCH PROJECT

Automatic Detection of Metastatic Breast Cancer Cells on Histopathological Slides of Lymph Nodes

Authors:
Charles BOY DE LA TOUR
Megan FILLION
Nicolas GREVET

Supervisors:
M. Stergios CHRISTODOULIDIS
Ms. Maria VAKALOPOULOU

October 2021 - April 2022

Our code is partially available at:
https://gitlab-research.centralesupelec.fr/charles.boy-de-la-tour/segmentation_wsi_breastcancer

A project carried out in
MICS Laboratory

as part of the Master of Sciences in Artificial Intelligence
MSc in AI

April 27, 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

As part of the Master of Sciences in Artificial Intelligence (2021-2022 promotion), we have been asked to do a research project for a period of 6 months, from the beginning of October 2021 to April 2022. We have joined the team of Stergios and Maria, members of the MICS Laboratory (MICS), in CentraleSupélec. The scope of this project was to investigate, implement and evaluate novel deep learning methods for the automatic processing of histopathological images (WSIs) in order to accurately and robustly detect metastatic tumor cells in lymph nodes.

According to the World Health Organization (WHO), breast cancer is the world's most prevalent cancer. In 2020, there were 2.3 million women diagnosed with breast cancer and 685 000 deaths globally and as of the end of 2020, there were 7.8 million women alive who were diagnosed with breast cancer in the past 5 years [1]. In the present research, histopathologic image analysis is the standard method applied in the clinical practice to diagnose breast cancer. Even though the prognosis for patients diagnosed with breast cancer is usually good, the survival rate declines if cancer metastasises [2]. That makes recognising the metastases in lymph node sections one of the most important prognostic factors.

In the process of histology image analysis for cancer diagnosis, pathologist visually observes the tissue, its distribution and regularities of cell shapes. After that process, pathologist decides whether there are some cancerous tissue regions and determines the malignancy level. However, this diagnostic procedure is time-consuming and small metastases are very difficult to detect even for experienced pathologists. Fortunately, computer based image analysis has become a rapidly expanding field within the past few years and whole-slide scanners are now commonly used for digitising glass slides at high resolution. This process partially allows automation of the histopathologic image analysis for cancer diagnosis, but there is still a great potential to improve and fully automate this task and help the pathologists to reduce their workload.

1.2 Goals

The main focus of this work is to develop a robust method for solving the task of the detection of metastases in whole-slide lymph node images using deep convolutional neural networks (DCNNs), based on previous works done by Stergios and Maria. To achieve that, it is necessary to get familiar with related work from the literature and

current state-of-the-art methods.

In the following chapters, a baseline solution for patches segmentation using DL will be created and tested on the data from the Gustave Roussy medical center. This technique will be improved, and patches will be aggregated to provide the full slide segmentation. The final solution will be compared with state-of-the-art methods regarding the performance, and proposed to Stergios and Maria for further explorations.

Chapter 2

Related works

2.1 CAMELYON challenges

In 2016, the CAMELYON challenge was proposed with the aim of developing computational pathology (CPATH) solutions for the detection of breast cancer metastases in sentinel lymph nodes. The introduction of the CAMELYON dataset was a game changer in the field of CPATH, as it made available for the first time the largest collection (n = 1,399) of fully manually annotated WSIs of sentinel lymph nodes of patients with breast cancer. It contained 399 WSIs for the CAMELYON16 and 1 000 WSIs for the CAMELYON17 and approximately three terabytes of image data [3]. This dataset was collected at five Dutch medical centres to ensure the slide heterogeneity.

Participants in the challenge had to solve two tasks designed to mimic routine tasks in pathology diagnostics: finding tumor regions in each lymph node and consequently predicting the presence of tumors at a WSI level. The impact of CAMELYON was similar in magnitude to that of ImageNet on the computer vision community. The large set of data and clinical focus of CAMELYON stimulated the creativity of both researchers and industry, who pushed forward the development of AI for metastasis detection, thereby enabling CPATH methods to make a leap from both academic and commercial technological perspectives.

PCAM is a huge, image classification dataset providing over 327 000 small patches of size 96 × 96 pixels extracted from the CAMELYON dataset to simplify the task of metastasis detection. Each patch is annotated with a binary label – a positive label indicates that the patch's central 32 × 32 pixel region contains at least one pixel of metastasis, a negative label indicates the opposite. If the tumour tissue is located in the outer region of the patch, it does not count as a positive label and it only provides additional information about the surrounding tissue [4]. Example of both positively and negatively labelled samples are illustrated in Figure 2.1.

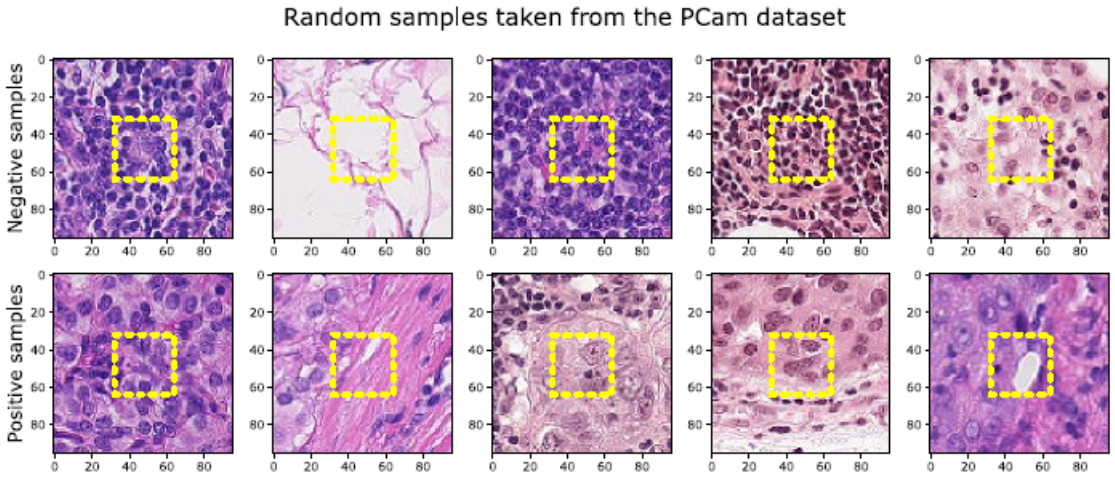

FIGURE 2.1: Randomly extracted patches with highlighted central 32 × 32 pixel region and both positive and negative labels from the PCam dataset.

For slide-based evaluation: algorithms are evaluated for their ability to discriminate every whole-slide image as either containing or lacking metastases and the ROC curve is used. For lesion-based evaluation: algorithms are evaluated for their ability to identify individual micro-metastases and macro-metastases in whole slide images and the free-response receiver operating characteristic (FROC) curve is used [5].

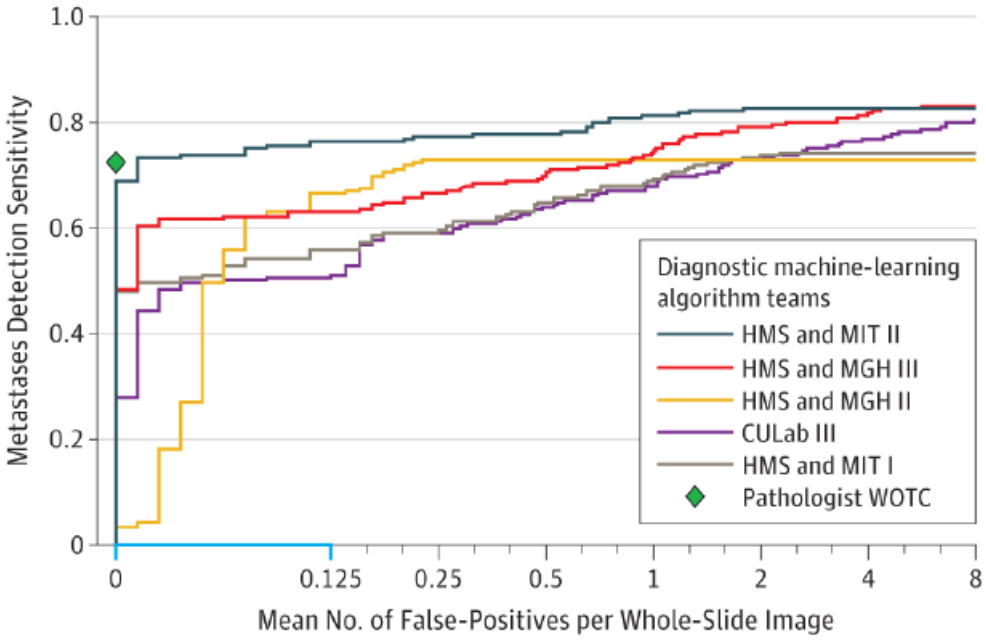

FIGURE 2.2: FROC curves of top five performing algorithms compared to pathologist without time constraint (WOTC) for metastases identification task

2.2 Literature review

2.2.1 Deep learning in histopathology: the path to the clinic

This is an article published in Nature Medicine in May 2021 and it was recommended by Stergios and Maria. It serves as a great review of all the current state of the field and is not overly complex on the subjects of digital pathology and deep learning. Also it describes the challenges that still need to be addressed before AI in histopathology can achieve clinical value.

Some of the main take-away points:

- CPATH automation could have a tremendous impact on the optimization of the clinical workload of pathologists by flagging suspicious regions or slides for inspection or, in the future, assessing cases autonomously.
- There is a lack of manual annotations in large-scale datasets. Clinical annotation of WSIs tends to be easier to achieve than manual annotation, but it is yet not very efficient to build CPATH on these type of data. AI need broad, widely used, and actually relevant clinical data and it is not possible today.
- Next to the challenge of collecting sufficiently large sets of annotated WSIs, prospective studies have to be conducted to show the true benefit of AI for histopathological diagnostics. Moreover, issues related to explainability, ethics and regulation are also insufficiently studied and will require more attention in the near future.

2.2.2 MANA

When we were first presented with this topic, we were told that cell size, shape, and spatial distribution are notable factors in a diagnosis of breast cancer in histopathological slides. Seeing as we didn't have masks delimiting nuclei regions within input patches, we decided to research an implementation that could automatically detect cell nuclei within an image and we found Salvi and Molinari's paper entitled 'Multi-tissue and multi-scale approach for nuclei segmentation in H&E stained images' [6]. Salvi and Molinari's motivation was to create an automated algorithm that can detect nuclei in histopathological slides in order to help pathologists notice early signs of cancer in their patient's tissue samples. Although this study was only done on hematoxylin and eosin stained stained samples, the algorithm reached state of the art results on different types of tissue and magnification.

The procedure of the MANA algorithm isn't necessarily computationally hard, but does require a number of subsequent transformations in order to achieve the final segmentation. In order to create a binary mask of cell nuclei, the researchers first convert the input to grayscale by projecting the input image to the HSL space and only retaining the luminance channel. They then compute the grayscale histogram where the number of bins is equal to the range of possible pixel values. Salvi and Molinari detail that potential thresholds for the nuclei masks could be found in the inflection points of the progressive weighted mean curve. The progressive weighted mean curve aims to extract relevant characteristics about the color distribution by computing the weighted mean of each bin up to date using this function.

$$PWM_{CURVE} = \frac{\sum_{i=0}^{P} w_i x_i}{\sum_{i=0}^{P} w_i}, \tag{2.1}$$

where w_i is the histogram count for the i^{th} class and x_i is the bin location. They then compute the polynomial fitted curve of the PWM to the 15^{th} degree and generate binary masks using the inflection points of the fitted curve as thresholds. Normally, 3-6 masks are generated in this step and the final threshold chosen is the one that yielded the highest median area of all detected objects.

In order to correct over and under segmentation errors, they calculate the mean, and not the median as detailed above, area of of all detected objects in a patch and classify each region into three classes: small (under 25% of mean area), large (equal or greater to five times larger than the mean), and anything in between as normal. They discard the undersegmentated regions because they assume these detected objects are too little to be a nucleus. The large regions are most likely a cluster of nuclei, on which MANA along with a marker-based watershed algorithm is recomputed on the sub-region in order to segment the cluster.

In order to measure the performance of the MANA algorithm, the researchers computed the recall, precision and F1-score based on whether a nucleus was correctly identified (as compared to a pixel wise metric). For example, the true positive represents the number of cells that were correctly identified but to what extent the nuclei segmentation was larger or smaller than the ground truth was not taken into account. They also defined a number of customized metrics, like CS, US, and SE which respectfully describe the number of identified nuclei that were correctly segmented, under segmented, or who had a segmentation error.

Organ	Computational time (s)	Object-level performance			Pixel-level performance		
		Recall	Precision	F1-score	CS (%)	US (%)	SE (%)
Colon	22.89 ± 2.15	0.9505 ± 0.0121	0.9048 ± 0.0114	0.9270 ± 0.0086	86.78± 1.97	8.69 ± 2.00	4.53 ± 1.14
Liver	11.32 ± 1.25	0.9249 ± 0.0267	0.9547 ± 0.0114	0.9392 ± 0.0101	87.17± 4.55	5.81 ± 2.42	7.02 ± 2.37
Bone	13.10 ± 1.13	0.9486 ± 0.0290	0.9362 ± 0.0203	0.9417 ± 0.0077	74.15± 4.17	21.35 ± 4.30	4.07 ± 2.49
Prostate	12.28± 1.31	0.9533 ± 0.0127	0.9404 ± 0.0147	0.9467 ± 0.0106	77.47± 7.72	18.79 ± 7.31	3.74 ± 0.88
Adrenal gland	18.02 ± 1.26	0.9126 ± 0.0300	0.9239 ± 0.0312	0.9174 ± 0.0129	84.60± 4.44	7.33 ± 2.55	8.06 ± 2.62
Thyroid	23.71 ± 5.94	0.9335 ± 0.0296	0.8914 ± 0.0221	0.9112 ± 0.0038	81.62± 6.18	12.61 ± 5.39	5.77 ± 2.60
Overall	16.89 ± 5.72	0.9372 ± 0.0288	0.9253 ± 0.0293	0.9305 ± 0.0161	81.97± 7.05	12.43 ± 7.32	5.53 ± 2.66

Data are reported as mean ± standard deviation

FIGURE 2.3: MANA results

We see here that MANA achieved very high object-level recall, precision and F1-scores, reaching its highest F1-score on Adrenal Gland tissue. In order to emphasize the performance of their newly developed algorithm, the research team decided to compare the results of MANA to widely used, open-source software such as CellProfiler, QuPath and Fiji. MANA outperformed all three algorithms, returning a f1-score that is 0.3 points higher than second best performer on this test set (Fiji). It should be noted that Fiji is semi-automated, meaning that the algorithm needs additional input from the user to run, while MANA is fully-automated.

Chapter 3

Data decription

3.1 Whole Slide Images

Histopathological images for breast cancer detection are an important tool used by pathologists to diagnose and monitor the progression of breast cancer. The images are created through a process of microscopic examination of tissue samples taken from a patient's breasts. The slides prepared from these tissue samples allow the pathologist to examine the cellular structure and morphology of the tumor in order to make a diagnosis.

The use of histopathological images for breast cancer detection has greatly improved the ability to detect and treat this disease. The images provide information about the size, shape, and location of tumors, which can help guide treatment decisions. Additionally, histopathological images can be used to monitor the response of tumors to treatment, allowing us to adjust treatments as needed.

To obtain these images, H&E staining is performed. It is a histological technique used to stain tissue samples for examination under a microscope. The sample is stained with hematoxylin, which colors nuclei blue, and eosin, which colors cytoplasm and other organelles pink. This combination of stains produces the characteristic H&E image with its characteristic purple-pink coloration.

Medical professionals are constantly seeking new and innovative ways to improve the quality of patient care. One such innovation is the use of whole slide images. Whole slide images are digital scans of entire glass slides, which can be magnified and manipulated to allow for a more in-depth examination of tissue samples. These WSIs (Whole Slide Images) are in the order of 10 Gb, which is too large to fit in a computer memory.

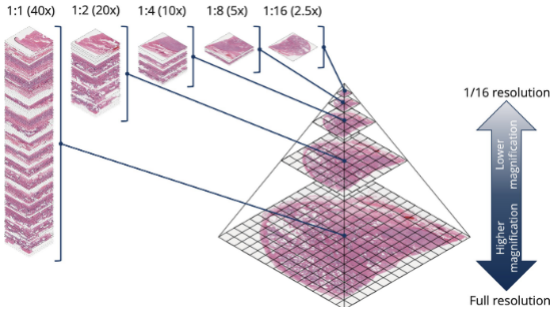

FIGURE 3.1: Patches extraction from different resolution levels.

Hence, after selecting a specific magnification, patches of 256x256 pixels are usually extracted. A classification or segmentation of these patches is performed.

In our project, 2 datasets have been used. A private one from Hospital Gustave Roussy (GR) which is held on MICS servers and a public one, CAMELYON.

CAMELYON: The CAMELYON breast cancer WSI dataset is a valuable resource for researchers studying the disease. 140612 patches have been extracted from this dataset and the luminosity has been standardized. Each patch has an associated label : 1 if cancerous, 0 otherwise.

GR: The Gustave Roussy cohort dataset consists of 70 positive WSIs from 153 patients with metastatic region annotation. A train, validation and test dataset have been created at the patient level. 236670 patches have been extracted from these datasets at level 10x, 20x, and 40x. The GR dataset has been mostly used in our approach due to the presence of the Ground Truth Mask at the patch level, allowing us to train segmentation models.

Chapter 4

Theoretical framework

4.1 Convolutional neural networks

4.1.1 UNet

First developed for medical imagery, the U-Net [7] architecture Figure 4.1 has proven fundamental for semantic segmentation as it built upon the FCN architecture proposed by Long et al. [8] by adding some skip connections which help keep track of spatial information that can get lost when downsampling through the network. Its fully convolutional nature enables it to take inputs from different sizes as the parameters are not – entirely – indexed on the input but rather on the number of layers.

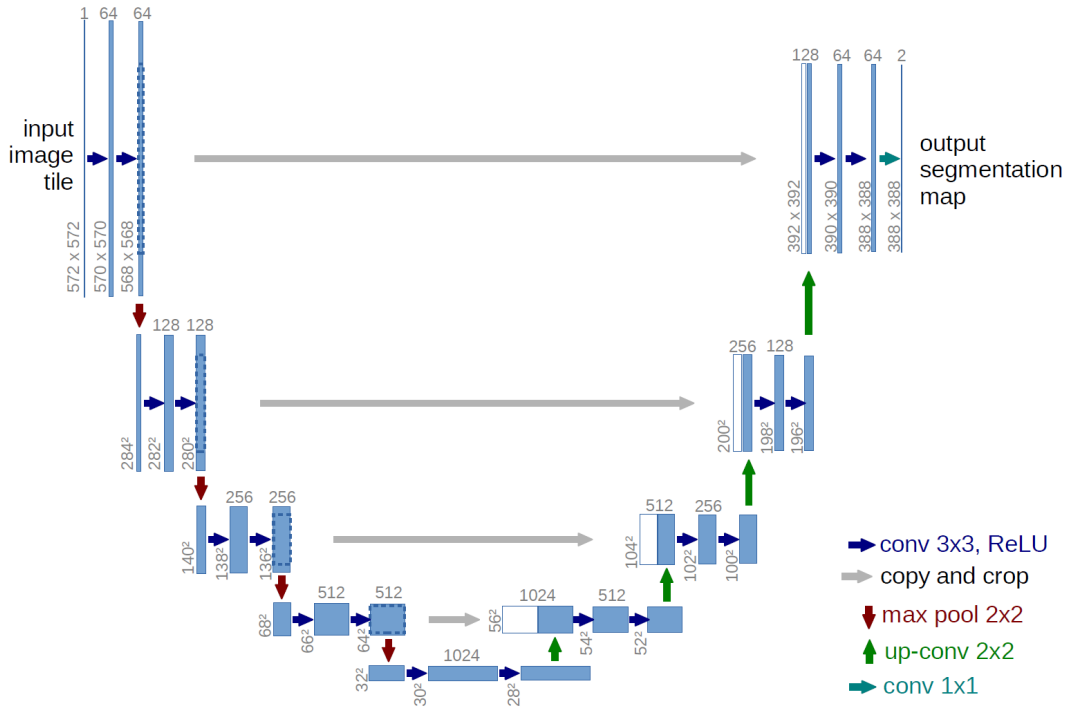

FIGURE 4.1: UNet architecture

4.1.2 UNet++

Unet++ [9] is a nested U-Net Architecture with varying depths Figure 4.2. It aims at improving the accuracy by including dense block and convolution layers between

the encoder and decoder. Hence, Unet++ complements the UNet architecture with two elements:

1. **Re-designed skip pathways:** The main idea is to redesign the skip pathways to bridge the semantic gap between the feature maps of the encoder and the decoder subpaths, through additional convolutions layers fusing semantically dissimilar feature maps. This may also impact the optimisation problem, making it easier for the optimiser to solve. They are inspired by DenseNet [10]

2. **Deep supervision:** It enables to decide the level of model pruning and thus the balance between speed (inference time) and performance of the model. The model can thus operate in two modes, the accurate mode where the outputs from all segmentation branches are averaged, and the fast mode where only one of the branches is selected.

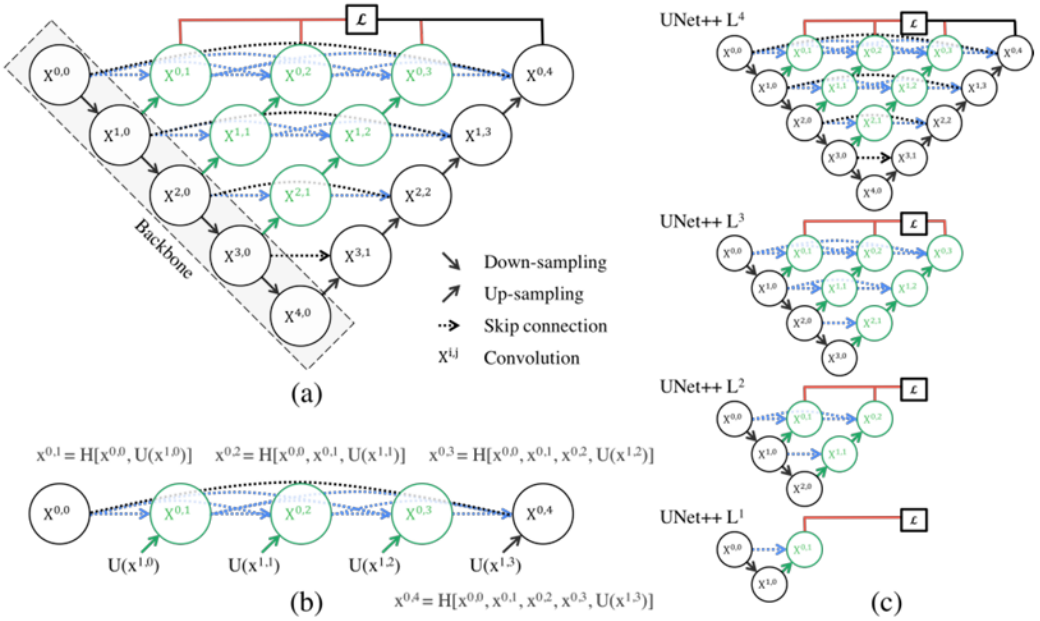

FIGURE 4.2: UNet++ architecture

4.1.3 DeepLabV3+

DeepLabv3+, which was invented by Google in 2018, combine the Encoder-Decoder architecture, still to recover the spatial information and a spatial pyramid pooling with atrous convolutions to encode multi-scale contextual information.

Atrous convolution, also called dilated convolution, is equivalent to convolving the input x with upsampled filters, which are produced by inserting d-1 zeros between two consecutive filter values along each spatial dimension. By adjusting d, we can adaptively modify filter's field-of-view. When d=1, it is standard convolution.

Atrous convolution allows us to enlarge the field of view of filters to incorporate larger context. It thus offers an efficient mechanism to control the field-of-view and finds the best trade-off between accurate localization (small field-of-view) and context assimilation (large field-of-view).

It is combined with a point-wise convolution and a depth-wise convolution and is included into a Spatial Pyramid Pooling as it is found that it significantly reduces the computation complexity of proposed model while maintaining similar (or better) performance

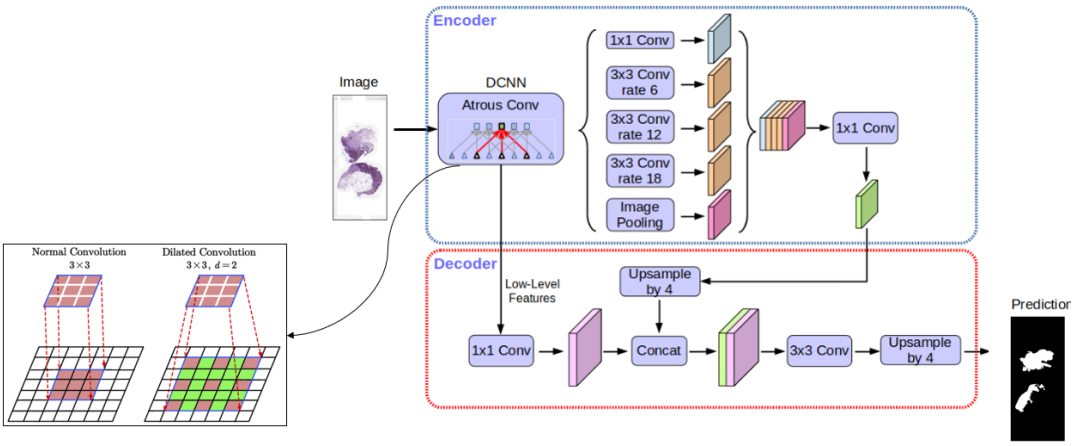

FIGURE 4.3: DeepLabV3+ architecture

4.2 Encoder architecture

Each of the used models has specified a so-called backbone. The term 'backbone' refers to a CNN performing the feature extraction. Rest of the segmentation framework is subsequently built around the extractor. Taking this into account, we can choose the most suitable backbone for our task.

EfficientNets is a family of novel model scaling method that uses a simple yet highly effective compound coefficient to scale up CNNs in a more structured manner. EfficientNets surpass state-of-the-art accuracy with up to 10x better efficiency (smaller and faster) [11]. We choose the same architecture, EfficientNet-B0, as the backbone for all three models, as it offers a good trade-off between computational cost and accuracy, as we can observe on Figure 4.4.

TABLE 4.1: Models complexity with EfficientNet-B0

Models	**Parameters**
UNet	100.000
UNet++	6.500.000
DeepLabV3+	4.500.000

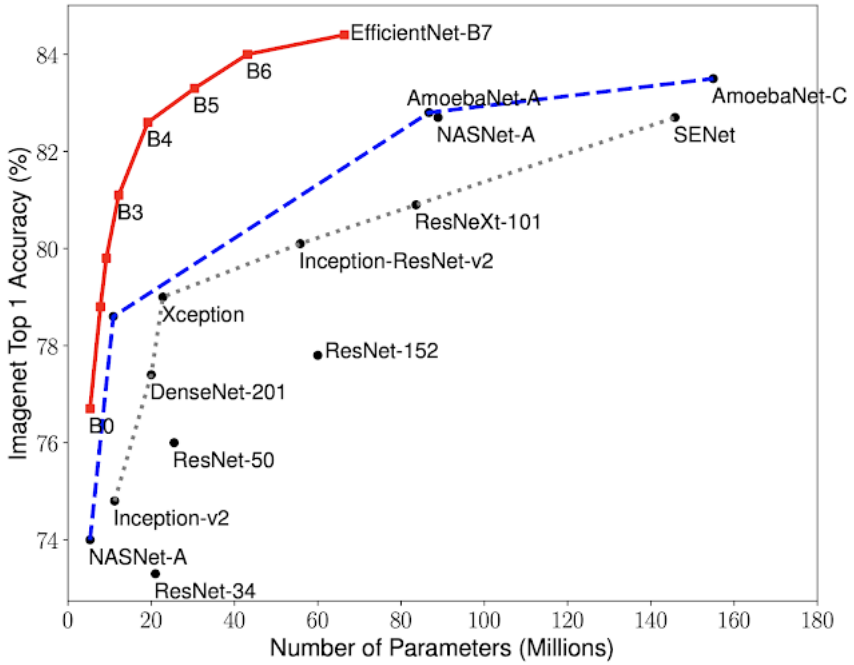

FIGURE 4.4: EfficientNets performance

4.3 Thresholds

Identifying tumors on the patches is a crucial step for the final algorithm efficiency.We use simple thresholding steps to transform the map into a binary one. We first use a simple thresholding algorithm at the pixel level: if the pixel's value is lower than a hand-crafted threshold, we convert it to the positive area of the prediction. Otherwise, we convert it to the negative area of the prediction. Doing so, we ensure that the extracted patches masks contain only two unique pixel values. We tried different values from 0.3 to 0.6 and we finally set 0.6 as it gives the best performance of the segmentation algorithm.

Then, we apply a second threshold at the patch level. To prevent our algorithm from generating too many FP's, we use a grid search to find an optimal threshold value for the minimal size of the area that should be taken as a proper tumour, not as a segment. The optimal value found is 0.15.

Chapter 5

Experiments and results

5.1 Segmentation pipeline

As we have already indicated, we moved the task from classifying patches as either containing metastasis or not to the segmentation of tumour regions on the patches. What that means is, instead of making a patch-level prediction, we make a prediction on a pixel-level. We aim to classify every pixel into one of two classes – containing tumour or tumour-free. That allows us to evaluate every slide on a pixel-wise level and bring the task of tumour localization closer to the clinical practice.

To accomplish the segmentation task, we propose to use some well-known CNNs designed for the semantic segmentation task : UNet, UNet++ and DeepLabV3+ models with a EfficientNet-B0 backbone. In this work, we do not use pre-trained versions of mentioned models.

5.1.1 Ablation study

In order to reach optimal scores, we went through a lengthy process of hyper-parameter tuning, trying different learning rates between 1e-3 and 1e-4 and weight decay value for regularization between 1e-4 and 1e-5. For our final training of all three models, we found that we reached optimal results on the test set with a learning rate of 5e-4 and a weight decay of 1e-4. Our models, especially UNet++ and DeepLabV3+ due to their complex structures, had a tendency to overfit so we tried to increase our batch sizes as much as possible. For UNet we implemented a batch size of 64 while for UNet++ and DeepLabV3+ we set the batch size to 32 – the highest possible size that our GPU could handle. We added early stopping to avoid overfitting, where the training stops after the validation and training loss diverge and we save the weights of the model at the epoch with the lowest validation loss. Each model was trained until convergence.

TABLE 5.1: Parameters selected and fine-tuned for the models performing the segmentation task.

Parameters	**UNet**	**UNet++**	**DeepLabV3+**
Learning rate	5e-4	5e-4	5e-4
Weight decay	1e-4	1e-4	1e-4
Batch size	64	32	32
Number of epochs	100	80	80
Early stopping		√	√

We also implemented a number of different losses, such as binary cross entropy and dice loss, where the Sørensen–Dice coefficient is used to compute the loss between the ground truth and the predicted masks. The binary cross entropy loss yielded higher results on the test set for all three models and therefore is the final loss function we settled on.

5.2 Segmentation with nuclei masks pipeline

Although we achieved high scores using the GR seg dataset for detection of breast cancer metastasis in WSIs, we wanted to see if adding additional input could increase the performance of our models. As mentioned in our related works, cell size, shape, and spatial distribution are notable factors in detecting breast cancer metastasis in a slide so we thought it would be interesting to add a binary mask delimiting the cell nuclei as an additional feature. As detailed in Figure 5.1, we first take the input patch and feed it into a UNet++ whose task is to output a binary nuclei mask. Then we concatenate the input patch, along with the generated mask, into a second UNet++ in order to output the final metastasis detection mask. We decided to use a UNet++ architecture for both the nuclei and tumor segmentation models as out of the three models trained for segmentation, it was the one that performed the best.

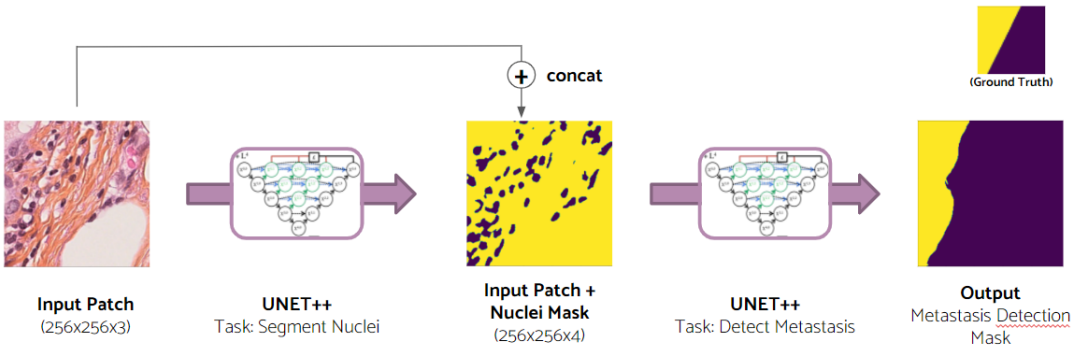

FIGURE 5.1: Diagram of metastasis detection with cell nuclei

5.2.1 Preprocessing

In order to train a model that can generate nuclei masks, we first had to generate ground truth masks delimiting the cell nuclei in our input patches. To do so, we used a combination of different methods at our disposal, notably the MANA algorithm detailed in our related works section and the better known Otsu algorithm which generates a binary mask using a threshold that maximizes inter-class variance. We made a couple of changes to the MANA algorithm, such as inverting the image. We chose to invert our input image before implementing the MANA algorithm because we found that the progressive weighted mean was more sensitive to pixel values closer to 255 and our cell nuclei tended to be on the lower end of that scale due to the dark nature of the staining. Also, the paper details that potential thresholds should be the inflection points of the second derivative of the fitted polynomial curve, i.e. where the derivative of the curve changes signs, but through our research we found that local maxima and minimas of the fitted curve also worked as potential thresholds.

This means that we had three algorithms to find potential thresholds – Otsu, MANA (where we compute thresholds from the first derivative of the fitted curve), and MANA prime (where computed thresholds were inflection points of the second derivative). We found that for magnification 10x, Otsu worked better than MANA to detect cell nuclei, but that being said the generated masks contained a lot of noise and there were a high number of outliers. For magnification 20x and 40x, MANA outperformed Otsu for almost every iteration. In general, MANA was more precise in its estimation of nuclei masks but tended to have a high number of false negatives while MANA prime was a little over-ambitious in its delimitation boundaries but has less false negatives.

For the pre-processing of our pipeline, we computed Otsu, MANA and MANA prime thresholds on every input patch in a dataset, and stored the mean and median values for each algorithm in a table found in Table 5.2. Then in our dataset loader, we'd take the mean and median values of each algorithm and compute the binary nuclei masks for each threshold. We'd then select the final threshold by keeping the value that yielded the highest median area of objects. We would then perform the post-processing detailed in related works in order to remove false positives and resegment clusters. The point of this step was to a) lower the computation time of the model by saving potential thresholds beforehand and b) trying to limit the number of ground truth masks that were completely wrong, also known as outliers. Seeing as nuclei were around the same pixel value in a dataset, this method proved to be coherent.

TABLE 5.2: Computed mean and median thresholds for Otsu, MANA and MANA prime on all three magnifications.

Dataset name	**Len dataset**	**Mean otsu**	**Median otsu**	**Mean mana**	**Median mana**	**Mean mana prime**	**Median mana prime**
X_train_10	7038	104.97	107.52	171.55	170	143.81	141
X_train_20	16859	114.75	116.75	172.21	171	143.34	142
X_train_40	28081	118.61	120.74	174.49	172	151	143

5.2.2 Nuclei segmentation model

For our nuclei segmentation model, we feed in our input patches of size 256x256 with three channels (RGB) into a UNet++ and generate the ground truth masks using the method detailed above. We used about the same configuration as the UNet++ model used for tumor detection, except seeing as nuclei detection is an easier task to learn, we lower the learning rate to 1e-4 and weight decay to 1e-5. For some of our tests, we tried changing the encoder and used a regnety architecture, which in a nutshell is an improvement on the original regnet architecture with additional parameters such as slide, quantization and attention. We found that the training took exponentially more time so we settled on an efficient net b0 encoder architecture for the final model. We use Adam optimiser, binary cross entropy loss and the model was trained until divergence. The output of this model is a binary mask delimiting nuclei of cells in a patch. The predictions of the trained model can be seen in Figure 5.2.

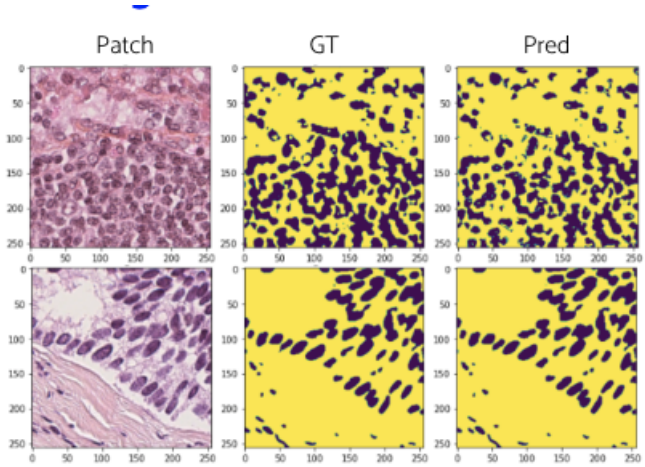

FIGURE 5.2: Input, ground truth and predictions of nuclei segmentation UNet++

5.2.3 UNet++ with nuclei mask

After having trained our model for nuclei segmentation, we now had to train our tumor segmentation algorithm. The preprocessing and data generation is identical to our classic segmentation model, the main difference lies in the forward pass function. We feed our input patch of size 256x256x3 into our pretrained UNet nuclei segmentation model, for which we have freezed the gradients, and then concatenate the input patch with the generated binary mask and feed the whole stack of size 256x256x4 into a UNet++. We varied the learning rate between 5e-4 and 1e-4, used a weight decay of 1e-5, Adam optimizer and binary cross entropy loss. It could have been interesting to train our nuclei and metastasis segmentation models at the same time and backpropagate the loss on both models, but we assumed that since one task was a lot easier than the other (nuclei detection) that the combined model would not learn as well. This could be an interesting modification to test out in future research. The output of the tumor segmentation model with nuclei mask is a binary mask delimiting the tumorous regions, as shown in Figure 5.3.

5.3 Evaluation metrics

In this section, we declare all the metrics used for the evaluation of trained models.

5.3.1 Pixel Accuracy

A simple metric to evaluate a segmentation task is to report the percent of pixels in the image which were correctly classified.

When considering the per-class pixel accuracy we're essentially evaluating a binary mask; a true positive represents a pixel that is correctly predicted to belong to the given class (according to the target mask) whereas a true negative represents a pixel that is correctly identified as not belonging to the given class.

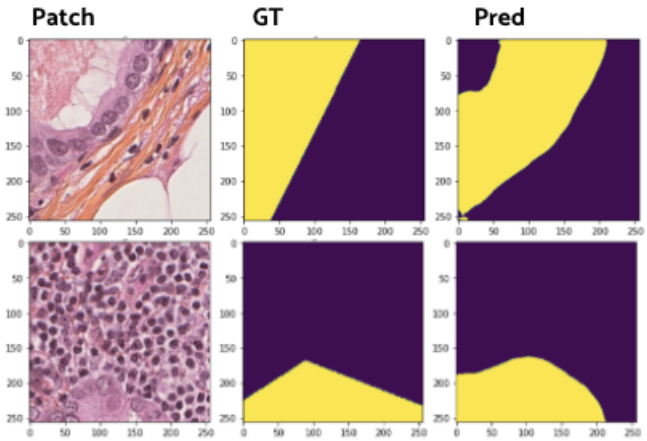

FIGURE 5.3: Input, ground truth and predictions of metastasis segmentation UNet++

This metric can sometimes provide misleading results when the class representation is small within the image, as the measure will be biased in mainly reporting how well you identify negative case (ie. where the class is not present).

5.3.2 Mean Intersection over Union

The Intersection over Union (IoU) metric, also referred to as the Jaccard index, is essentially a method to quantify the percent overlap between the ground truth mask and our prediction output. Quite simply, the IoU metric measures the number of pixels common between the ground truth (G) and prediction masks (S) divided by the total number of pixels present across both masks.

$$IoU(G,S) = \frac{|S \cap G|}{|S \cup G|}, \tag{5.1}$$

The intersection is comprised of the pixels found in both the prediction mask and the ground truth mask, whereas the union is simply comprised of all pixels found in either the prediction or target mask.

The IoU score is calculated for each class separately and then averaged over all classes to provide a global, mean IoU score of our segmentation prediction.

5.3.3 Dice Similarity Coefficient

The Dice Similarity Coefficient (DSC) measures the degree of overlap between the segmentation region and ground truth, and is defined as

$$DSC(G,S) = \frac{2|S \cap G|}{|S| + |G|}, \tag{5.2}$$

where S is the predicted mask and G is the corresponding ground truth mask. DSC values range from 0 to 1, a higher value representing a better segmentation.

This metric is widely use in segmentation task and closely related to the IoU metric.

5.4 Analysis of the solutions

5.4.1 Results for the pixel-level segmentation

Since in later epochs our models may be overfitted, we examine results of the whole training process and try to find an epoch with the best performance.

Detailed results of this analysis for all three models are described in detail in Table 5.2. From the results summarised in this table, it is evident that we achieve the best results in all aspects with the UNet++ with a EfficientNet-B0 backbone and we decide to use it in the rest of the pipeline. Figure 5.x illustrates some random samples of patch-level predictions performed by the UNet++ model.

TABLE 5.3: Results of the CNN training processes for all three models at the pixel level for patches from the 10x magnification

Metrics	**UNet**	**UNet++**	**DeepLabV3+**
Pixel Accuracy	0,800	**0,873**	0,850
mIoU	0,664	**0,760**	0,720
DSC	0,835	**0,884**	0,852

5.4.2 Results for the patch-level segmentation

TABLE 5.4: Results of the CNN training processes for all three models at the patch level for patches from the 10x magnification

Metrics	**UNet**	**UNet++**	**DeepLabV3+**
Patch Accuracy	0,842	**0,882**	0,855
False Positives Rate	0,202	0,095	**0,083**

5.4.3 Results for the segmentation with nuclei masks

For the nuclei segmentation pipeline, we trained a UNet++ for nuclei detection and a second UNet++ that takes in an input patch and a nuclei mask in order to predict a metastasis detection segmentation mask. Unfortunately, the MANA and Otsu algorithms didn't work as well on 10x magnification so we decided to implement the segmentation with nuclei pipeline on 20x magnification, seeing as our previous models performed better on 20x magnification than 40x.

The nuclei segmentation model works astoundingly well, yielding a dice score close to 1. But, the model also seems to learn the noise present in the training set and has trouble segmenting individual nuclei in a cluster.

TABLE 5.5: Results of nuclei segmentation model

Model	**Pixel accuracy**	**Pixel recall**	**Pixel precision**	**DSC**
Nuclei segmentation	0.956	0.987	0.956	0.998

As for our segmentation model that takes four channels as input (RGB and binary nuclei mask), we improve the performance of the UNet++ scores trained on 20x magnification for all metrics except for false positive rate, as we can see in table 5.6. The performance boost for the segmentation model with additional input was not as high as we had hoped.

TABLE 5.6: Results of segmentation model with nuclei mask as input

Metrics	**UNet++**	**UNet++ with nuclei mask**
Patch Accuracy	0,859	**0,862**
False Positives Rate	**0,146**	0,174
False Negatives Rate	0,131	**0,108**
Pixel Accuracy	0.854	**0.855**
DSC	0.837	**0.855**

Chapter 6

Conclusion

Throughout this project, we were able to implement a successful segmentation pipeline in order to detect breast cancer metastasis in lymph node slides. We tested a number of different architectures such as UNet, UNet++ and DeepLabV3+. We even took it a step further by engineering a script that can split a WSI into patches, run it through our pretrained models, and then generate a whole slide mask highlighting the regions that were predicted to contain tumorous cells. We also implemented a pipeline that can predict a nuclei segmentation mask in order to further improve our tumor detection segmentation model performance.

Although we were thorough in the training and testing of our models and put in the time to explore additional paths to visualize and improve our predictions, there are still a number of modifications one can implement to this project. For starters, each of our models can only take one magnification as input. We trained and tested all three of our models in our original pipeline on 10x, 20x and 40x while our models in the segmentation with nuclei mask pipeline are only trained on 20x. It could be interesting to engineer a multi-scale approach, either by training separate models for each magnification and concatenating the results on a whole slide in order to reach the most accurate prediction or create a model that is scale invariant, where you feed an input patch along with its magnification to the model in the hopes of predicting tumorous regions.

Within the context of the segmentation with nuclei mask pipeline, another possible alternative that could yield higher results would be to add attention to the metastasis detection model. For the moment, we add the generated nuclei mask as a fourth channel to the input, while it could be interesting to feed the binary mask as an attention layer, as one would do in an NLP task in order to orient the model towards important regions.

References

[1] World Heath Organization. *Breast cancer*. URL: https://www.who.int/news-room/fact-sheets/detail/breast-cancer.

[2] Howlader N, Noone AM, Krapcho M, Miller D, Brest A, Yu M, Ruhl J, Tatalovich Z, Mariotto A, Lewis DR, Chen HS, Feuer EJ, Cronin KA (eds). *SEER Cancer Statistics Review, 1975-2018, National Cancer Institute*. URL: https://seer.cancer.gov/archive/csr/1975_2018/.

[3] G. Litjens, P. Bandi, B. E. Bejnordi, O. Geessink, M. Balkenhol, P. Bult, A. Halilovic, M. Hermsen, R. van de Loo, R. Vogels, Q. F. Manson, N. Stathonikos, A. Baidoshvili, P. van Diest, C. Wauters, M. van Dijk, and J. van der Laak. *1399 HE-stained sentinel lymph node sections of breast cancer patients: The CAMELYON dataset*. Vol. 7. 2018.

[4] B. S. Veeling. *GitHub - basveeling/pcam: The PatchCamelyon (PCam) deep learning classification benchmark.* URL: https://github.com/basveeling/pcam.

[5] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. Van Der Laak, M. Hermsen, Q. F. Manson, M. Balkenhol, O. Geessink, N. Stathonikos, M. C. Van Dijk, P. Bult, F. Beca, A. H. Beck, D. Wang, A. Khosla, R. Gargeya, H. Irshad, A. Zhong, Q. Dou, Q. Li, H. Chen, H. J. Lin, P. A. Heng, C. Haß, E. Bruni, Q. Wong, U. Halici, M. Ü. Öner, R. Cetin-Atalay, M. Berseth, V. Khvatkov, A. Vylegzhanin, O. Kraus, M. Shaban, N. Rajpoot, R. Awan, K. Sirinukunwattana, T. Qaiser, Y. W. Tsang, D. Tellez, J. Annuscheit, P. Hufnagl, M. Valkonen, K. Kartasalo, L. Latonen, P. Ruusuvuori, K. Liimatainen, S. Albarqouni, B. Mungal, A. George, S. Demirci, N. Navab, S. Watanabe, S. Seno, Y. Takenaka, H. Matsuda, H. A. Phoulady, V. Kovalev, A. Kalinovsky, V. Liauchuk, G. Bueno, M. M. Fernandez-Carrobles, I. Serrano, O. Deniz, D. Racoceanu, and R. Venâncio. *Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer*. Vol. 318. 2017.

[6] M. Salvi, N. Michielli and F. Molinari. *Stain Color Adaptive Normalization (SCAN) algorithm: separation and standardization of histological stains in digital pathology.* URL: https://www.sciencedirect.com/science/article/pii/S0169260720305721?casa_token=RgHOG5vA-MwAA%20AAA:WfCKz5AlwgzWDI52gANRHikmCchfE60MOvSWk5qoc_86dsb-i_8mjmGhEar0qL5rd-fvv7KK.

[7] Xiaomeng Li et al. "H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes". In: *IEEE transactions on medical imaging* 37.12 (2018), pp. 2663–2674.

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

[9] Zongwei Zhou et al. "Unet++: A nested u-net architecture for medical image segmentation". In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[10] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[11] Google AI Blog. *EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling*. URL: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html.