CANCERS DIAGNOSIS PROJECT WITH CNN [CANCERAPY APPLICATION]

Abdelrahman Karawia
Department of Computer Science
Mansoura University, FCIS
Mansoura

Technical Report

May 5, 2022

Abstract: This project is going to consider more than **three cancers** and tumors like breast tumor, skin cancer, and colorectal tumor for now and each model of them will aim to detect or to be more specific to classify the given data like an image (Radiology) which will be given as an input to the model which will be deployed as a mobile application and classify the cancer, or the tumor based on their different types. The problem of tumor classification for its malignancy level and the cancer for its type has been studied for the last few years and has a reasonable number of applications here and there for everyone who's trying to improve a model which has a reasonable level of predictability. There are enormous number of algorithms to be used in machine learning for such a classification problem and this would a great challenge to choose between, not only this we have also to figure out which hyperparameters for every model we are trying to deploy (Tuning Process). There are different number of algorithms and architectures to be used when dealing with neural networks models, starting with **conventional** neural network model with specific number of layers and nodes or units and apply different techniques to get the best test accuracy but as we are dealing with images (Computer Vision Model), it's better to use **CNN** which is a class of ANN as well, and it's predictable to be more accurate and less computation time as well. All of these models are modeled by the aid of TensorFlow framework for training and evaluating process, also it will be very effective for Converting the best model for each cancer or tumor to a TensorFlow Lite model to be deployed as mobile application using Flutter framework.

Keywords: Machine Learning, Computer Vision, Image Classification, Binary Classification, Multi-Class Classification Tuning, CNN, Neural Network, ANN, TensorFlow, Keras, Flutter, dart.

Table of Contents

Figures and Tables

1. MOTIVATION

Cancer is a **leading** cause of death worldwide, accounting for nearly 10 million deaths in 2020, or nearly one in six deaths [1], but the patient can handle that cancer or at least limit its symptoms with early detection which is the main key here.

The number of cases for every cancer or tumor changes significantly but its not all about the number of each case yearly, it's about the death rate for each one as well. According to [1], we are going to introduce the most dangerous cancers and tumors.

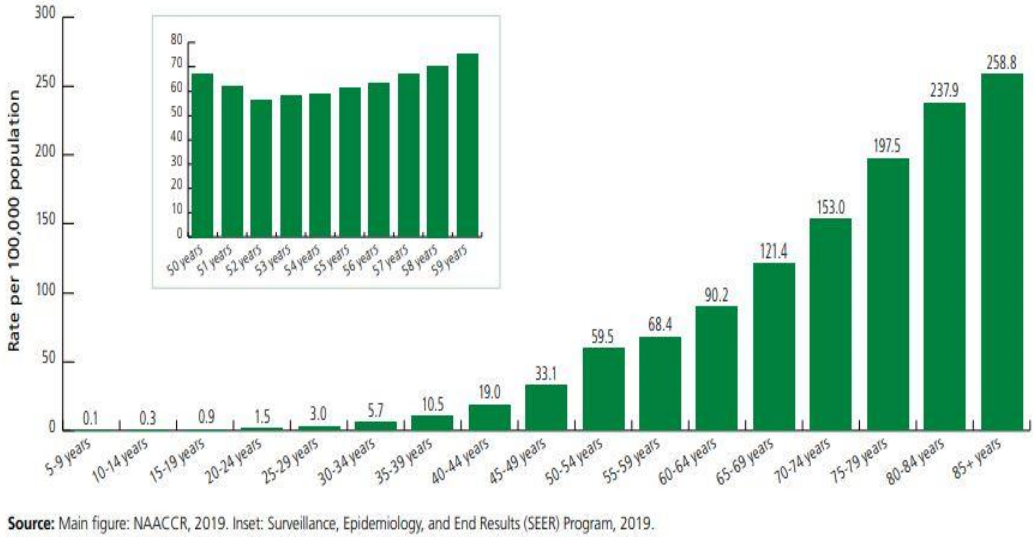
Table 1. Cancer New Cases and Deaths

New Cases		**Cancer Deaths**	
Breast	2.26 million	Lung	1.80 million
Lung	2.21 million	**Colon and Rectum**	916,000 deaths
Colon and Rectum	1.93 million	Liver & Stomach	900,000 deaths
Skin	1.2 million	**Breast**	685,000 deaths

First version of that project is including three types of cancers which has been bolded above in Table 1, which are Breast, Colorectal (Colon and Rectum Combined), and Skin Cancer and as time moves on, the project will hold most of known cancers with a reasonable dataset to be trained.

Breast Cancer is considered as the most common type of cancer between females and the first cause of cancer death among them In Egypt, it constitutes 33% of female cancer cases and more than 22,000 new cases diagnosed each year. This is expected to rise **exponentially** over the next years given the enlarging population [2]. **Early Detection and Diagnosis**, it's not just a combination of two words it's the key for treatment and the only way to be far away from the danger zone [3].

Approximately 4.4% of men (1 in 23) and 4.1% of women (1 in 25) will be diagnosed with **CRC** (Colon and Rectal Cancer) in their lifetime [4]. Lifetime risk is similar in men and women despite higher incidence rates in men because women have longer life expectancy. Age feature had a huge influence on CRC rates between population. The following figure is showing a left skewed distribution for rates of CRC with age.



Source: Main figure: NAACCR, 2019. Inset: Surveillance, Epidemiology, and End Results (SEER) Program, 2019.

Figure 1. CRC Rates with Age

Skin Cancer as well is one of the most dangerous cancers as most people at first stages ignores any symptoms on skin considering it as a normal lesion. Skin cancer is challenging case to be studied as it has large ratio of **misdiagnosed** cases as it contains multi classes to classify between and they are very close to each other, also the one type of them has dozens of shapes depends on diameter of lesions, the color as well, … etc.

Therefore, it's important to develop an effective classifier for each cancer to detect or classify its type with the aid of machine learning. It could be of great help to automate the detection and locate tumor tissue cells and to speed up the process. This way one would be able to overcome the dependence on the pathologist which would be especially useful in regions where no experts are available but that's not the main aim of the project which will be discussed latter.

2. INOTRODUCTION

Machine learning algorithms with just few years after putting the eyes of interest on it, exponentially increased in terms of applications involved in and the accuracy of detecting as well. Computer vision is a great topic in machine learning which is simply machine with an eye, so if we are talking about images with tracing something on it or detecting which class the image belongs to, all of this is under the term computer vision.

Therefore, our project is considered as a computer vision application by training a large set of images of specific cancer as input, then deploy the model which will predict the type of that cancer immediately with reasonable percent of accuracy.

Neural Network is the main algorithm to be working on, images trained by taking each pixel as a single unit of input considering we have only one channel for simplicity. So, if we are talking about (200, 200) resolution by RGB mode three channels involved then we have about 120,000 features for the input layer only, also nowadays we have a variety and large number of datasets as every institution is trying hard to help with improving the diagnosis with computers.

All the above arguments makes classical machine learning algorithms fades in some applications, and the neural networks are leading and improving day by day. Figure 2 shows by a simple figure the concept of what we talked about.

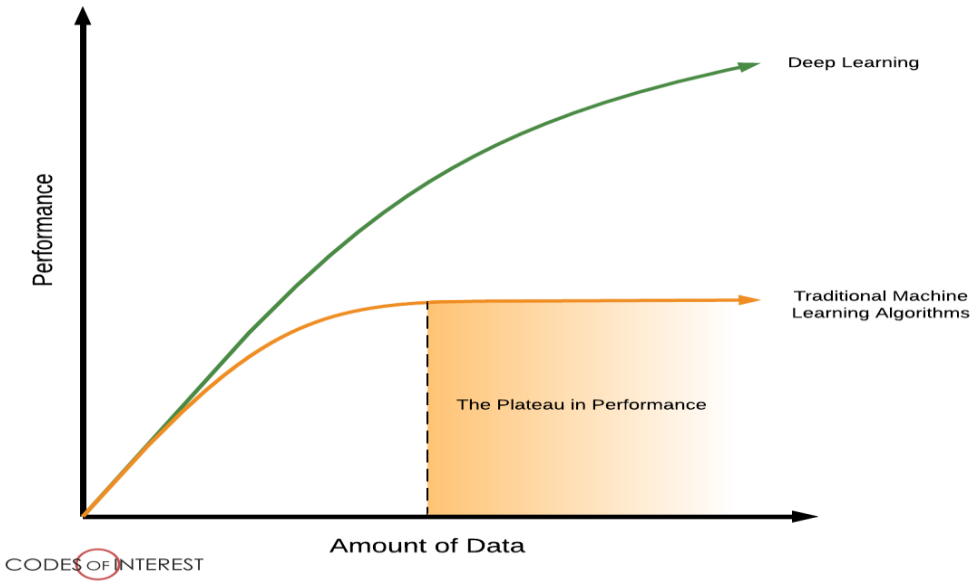


Figure 2. Performance of Classical ML and Deep Learning with Data

The Project (Cancers Classifier) Main goal is not just to help people at wild and abandoned areas, but hopefully deploying a versatile and large system with most of cancers on it to be deployed in every medical institution by uploading the radiology as usual but the system will automatically predict the cancer type or it's malignancy level and report all of that on the patient medical record and label it as a machine guess.

But for the first version of the project, it will not be embedded at any institution, it will just be deployed in a mobile application with quite good UI with flutter framework.

3. Intelligent Agent

3.1 PEAS

Table 2. PEAS Details

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical Diagnosis System	Healthy Patient, Minimize Costs, High Confidence	Patient, hospital, staff	Screen Display (Medical Questions, Images uploading, diagnoses, results)	Clicks (Images), Keyboard (Patient Info.)

3.2 ODESDA

Table 3. ODESDA Details

Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Medical Diagnosis System	Partially	Single	Stochastic	Sequential	Dynamic	Continuous

3.3 State-Space Problem Formulation

- **States:** The state exist in our model can be the parameters settings and configuration of them.
- **Initial State:** These states which are parameters settings and configurations have values at first all random.
- **Actions:** Moving around in parameter space.
- **Goal Test:** The project seeks to accomplish a high training accuracy compared to real life estimations.

- **Path Cost:** With the set of actions we have, and optimization done to get the optimal solution , we have cost associated with them, and for our problem it's the time taken to find the optimal solution.

4. METHODS

4.1 Datasets

4.1.1 Breast Cancer Dataset

Our model trained on a dataset of **Mammography** images of INbreast database was originally collected from Centro Hospitalar de S. Joao [CHSJ], Breast center, Porto. INbreast database collects data from Aug. 2008 to July 2010, which contains 115 cases with a total of 410 images and becomes **7632 images** with augmentation process [5].

This dataset not just consist of 2 classes benign and malignant, but also the masses of dense in this mammography image of the breast, and the breast density is divided into four categories according to BI-RADS standards, which are Entirely fat (Density 1), Scattered fibro glandular densities (Density 2), Heterogeneously dense (Density 3), and extremely dense (Density 4), therefore we have 8 classes to classify between as shown in Table 4.

Table 4. Breast Cancer Classes

Benign	Malignant
Density 1	**Density** 1
Density 2	**Density 2**
Density 3	**Density 3**
Density 4	**Density 4**

4.1.2 Skin Cancer Dataset

Harvard Data verse introduces one of the biggest datasets for a certain cancer or tumor which is HAM10000 ("Human Against Machine with 10000 training images"), data is introduced as **Dermatoscopic** images from different populations, acquired and stored by different modalities [6].

Our model will train on three different classes of skin cancer which are melanoma (**Mel**), melanocytic nevi (**Nv**), and benign keratosis-like lesions (**bkl**), but unfortunately the three types of cancer are not balanced which will affect the prediction process as it is going to be biased, so we have mission here to do with data before processing.

4.1.3 Colorectal Cancer Dataset

This data set represents a collection of textures in histological images of human colorectal cancer. All images are RGB, 0.495 µm per pixel, digitized with an Aperio ScanScope (Aperio/Leica biosystems), magnification 20x. Histological samples are fully anonymized images of formalin-fixed paraffin-embedded human colorectal adenocarcinomas (primary tumors) from our pathology archive (Institute of Pathology, University Medical Center Mannheim, Heidelberg University, Mannheim, Germany) [7].

This dataset consists of 5000 images for **8 classes** 625 image (150, 150) px each class (Balanced), we got Adipose Tissue, Complex Stroma, Debris and Mucus, Empty, Immune Cell Conglomerates, Mucosal Glands, Simple Stroma, and Tumor Epithelium.

4.2 Models

For every dataset listed before is going to be a input for neural network, the neural network may be plain (conventional) or convolutional (CNN) and for CNN we are going to use custom models once, known architectures as well like ResNet101, or MobileNetv2 with transfer learning concept.

The key difference between plain NN or Connected layers NN and CNN, is that for plain NN as shown in Figure 3, the input pass through each unit in the next layer by two steps one is linear step and the second is adding some non-linearity to the system (activation function) and the number of input in the previous layer is equal to the number of parameters of that node which are going to be trained, therefore it has a huge number of parameters to train.

For CNN Figure 4, each layer is a filter and the earlier layers is more simple than later layers which has the complex part of the image filtered, but both CNN and plain NN has the same concept of training to some point.

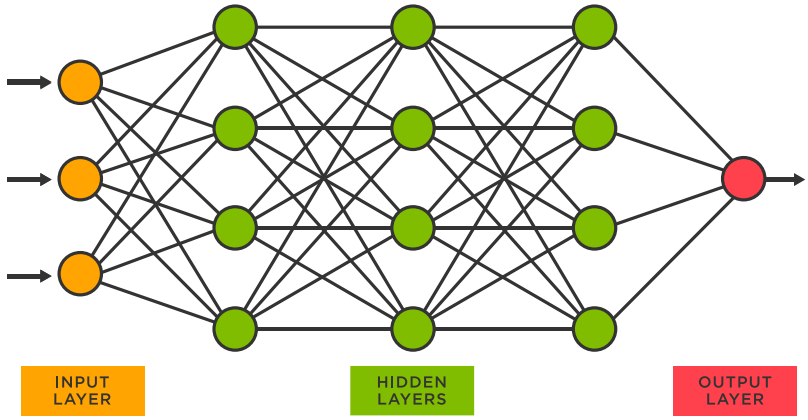


Figure 3. Plain Neural Network

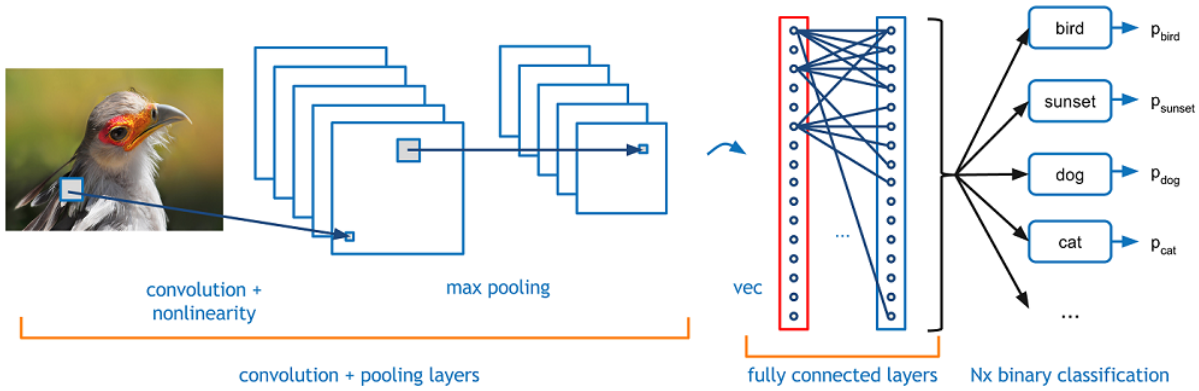


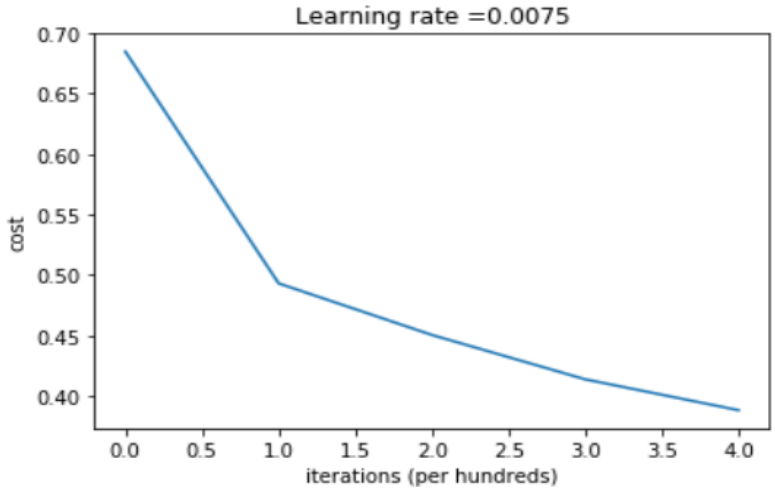
Figure 4. Convolutional Neural Network

4.2.1 Breast Cancer Model

First Model trained was a plain Neural Network model which coded from scratch without the use of TensorFlow framework on a small dataset was found at first and it was binary classification with only two classes malignant or benign.

With 510 training examples and 136 test examples used, and with 200*200 px with 3 channels we got 120,000 inputs, by using 7 nodes for the hidden layer and 2000 iterations. The hidden layer used RELU activation layer and sigmoid for output layer to give probability between 0 and 1.

We got an overfitted model with train accuracy about 99.8% and 66% test accuracy which is seems to have a high bias, so we got two options now to eliminate that overfitting by increasing the number of examples of the training set or apply regularization like dropout technique for instance, which decreasing the model sensitivity. Also, we can edit the stopping criteria as a quick solution.



To save lots of computation time we will move into CNN model directly using MobileNetv2 Architecture which is a great for low computation device like mobile phones. Using this pretrained model with its weights from ImageNet dataset and apply to our model for the same dataset we used in the previous model trained with the aid of **TensorFlow framework** as shown in Figure 5.

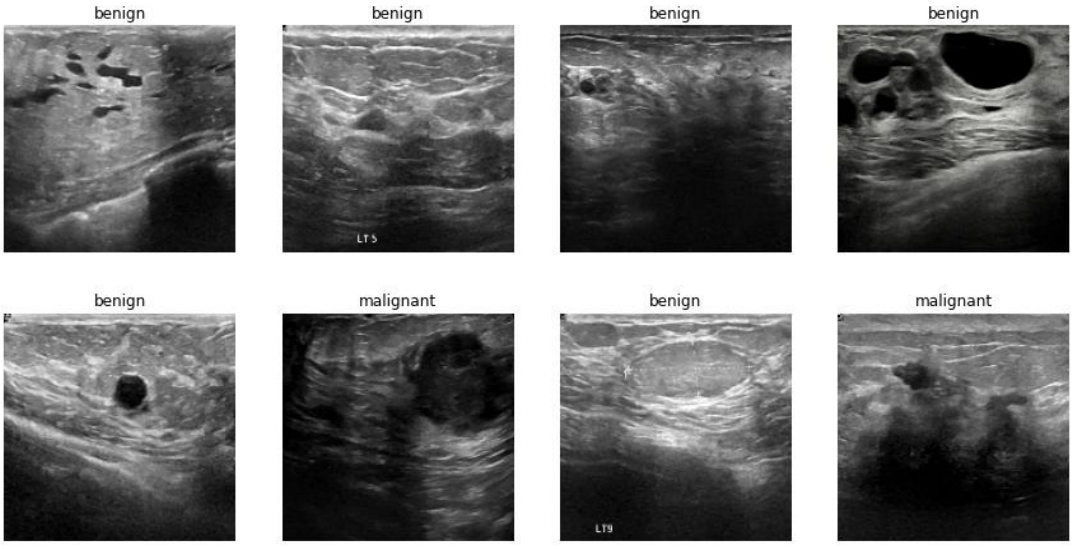


Figure 5. Sample of Dataset Breast Cancer (Authentic)

Using **Augmentation** will be a great help as well by creating a additional copies of the images but with different flips and orientations.

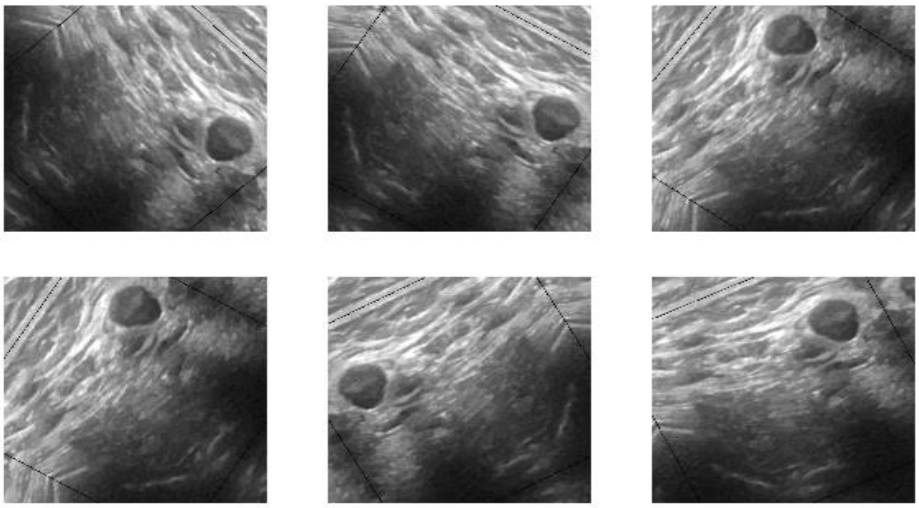


Figure 6. Images Augmentation (Authentic)

We will not include the top layer of MobileNetv2 and for exchange we will add two layers Average pooling2D and linear function for output layer. **Dropout Technique** will be used as well by 0.25 probability of dropping out (Regularization).

For first 5 epochs we will use weights of MobileNetv2 as it was from ImageNet dataset, we got accuracy 91% for training and 80% for test set.

Then we will be fining the model by make some of last layers of MobileNetv2 to be trainable and the last layers having the most parameters though. Then after 5 more epochs. We got **94%** training accuracy and **83%** test accuracy which is much better than before, and all of this takes small amount of time.

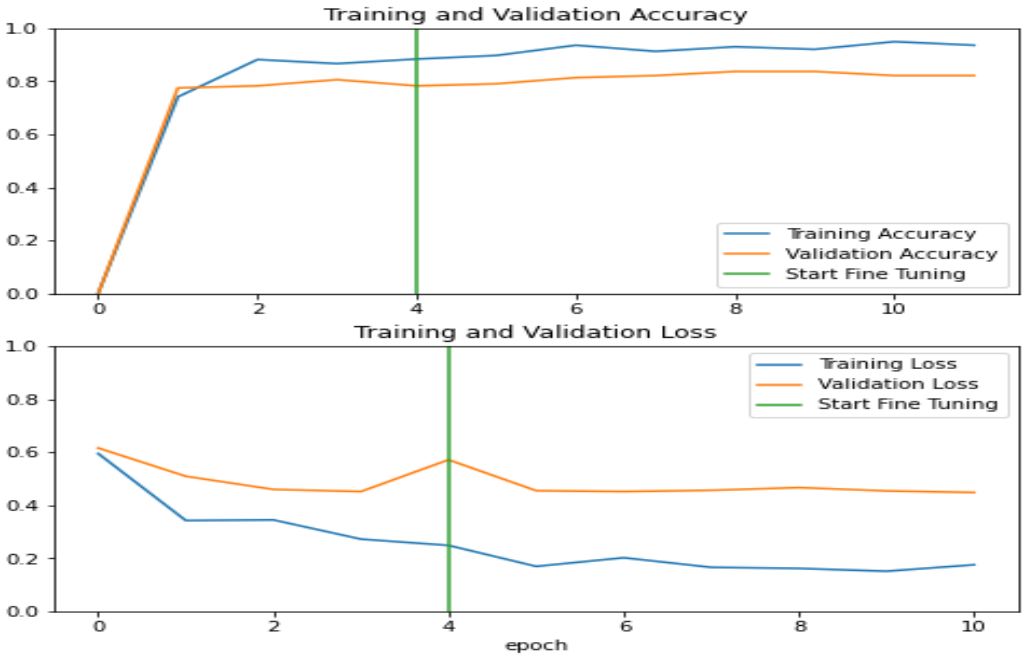


Figure 7. Accuracy and Loss with epochs (Converge)

Using the main dataset which mentioned before in Dataset Section, which contains about 7800 images, 6000 of them to be trained. But this time converting to ResNet101 architecture, and applying the same procedure exactly as MobileNetv2, this time we got eight classes benign and malignant plus **masses of densities** as shown in Figure 8.

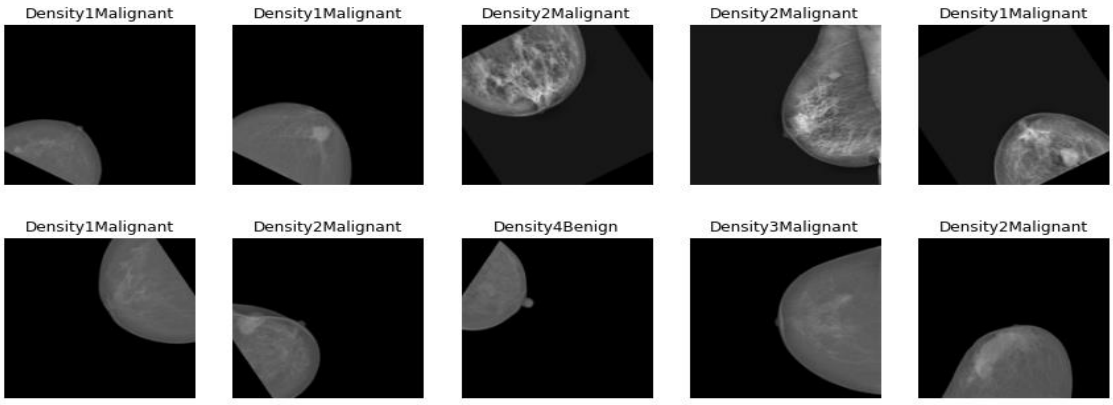


Figure 8. Breast Cancer Classes with masses (Authentic)

Dense breast tissue is detected on a mammogram [8], dense breasts indicates increasing in the chance that breast cancer may go **undetected by a mammogram**, since dense breast tissue can mask a potential cancer and increasing the risk of getting breast cancer. Therefore, even if the model indicated a benign for the patient but with high dense, so extra tests for the patient is mandatory to ensure that the benign class is benign and high dense of the breast not giving a false positive. We got an **overall accuracy** for the eight classes which is much harder as a model than above to classify:

$$Training\ Acc = 92\%, \quad Test\ Acc = 85\%$$

How can we evaluate our model, or what's the above test accuracy gives us actually? , what we can really do is to compare it with real life (compare apples to apples) to figure out the limitation of the model. Overall, the **sensitivity** of mammography is about **87 percent** [9]. This means mammography correctly identifies about 87 percent of women who truly have breast cancer in real life. So according to our 85 percent for 8 classes classification, we are doing great.

Figure 9, shows a simple code for developing random take of the validation set every time the code is running and evaluate the image described (It's Working well).

```
for imgs,labels in validation_dataset.take(1):
    plt.imshow(imgs[11].numpy().astype("uint8"))
    detect = list(labels[11].numpy())
    print('The truth ground class is ' + class_names[detect.index(max(detect))])
    img = tf.keras.preprocessing.image.img_to_array(imgs[11])
    img = np.expand_dims(img, axis = 0)
    predict = reconstructed_model.predict(img)
    predict = list(predict[0])
    print('The predicted class is ' + class_names[predict.index(max(predict))])
```

```
The truth ground class is Density3Malignant
The predicted class is Density3Malignant
```

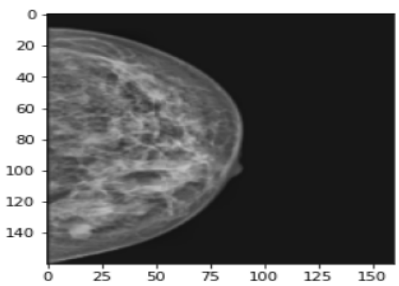


Figure 9. Evaluating Breast Cancer Multiclass Model (Authentic)

If we eliminated the masses from dataset and do the same model with appropriate hyperparameters and the output will be only benign or malignant using ResNet101 as well, we got high performance model with:

$$Training\ Acc = 98.6\%, \quad Test\ Acc = 96.7$$

Figure 10. Breast Cancer Binary Class Classifier Accuracy and Loss (Authentic)

Something to be noticed, that we just preprocess the dataset we have and put it there in the model to run and get the accuracy, we moved through tens of models with idea and experiment it and get the results.

The idea is how to tune the hyperparameters you have for instance choosing the right optimizer for your model like Adam Algorithm or specifying the best layer from pretrained model to make trainable or deciding which loss function to use based on the nature of the model we are working on, all of that is very exhausting process but necessarily.

Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can doing well with noisy problems, after missing around here and there by different algorithms Adam is the choice for every model we trained already, and this might be the only property or hyperparameter won't change with models.

4.2.2 Skin Cancer Model

Here we come to the most interesting model, which is skin cancer model, there a research were made by Heidi M Rolfe [10], for evaluating the accuracy in skin cancer diagnosis as accurate classification of the malignancy of the skin lesion is very vital to avoid mortality but on the other hands we have to take into consideration not taking a biopsy of a benign lesion to avoid high costs.

We are working on three classes for skin cancer to be classified by our model, one is malignant (**mel**) and the other two are benign (**nv** and **bkl**). But these data are not equal in terms of number of images or not even close as shown in Table 5 and Figure 11.

Table 5. Number of Images for Skin Cancer Types

Number of Images for each Class	
Melanocytic Nevi (nv)	6705 Images
Melanoma (mel)	1113 Images
Benign lesions of the keratosis (bkl)	1099 Images

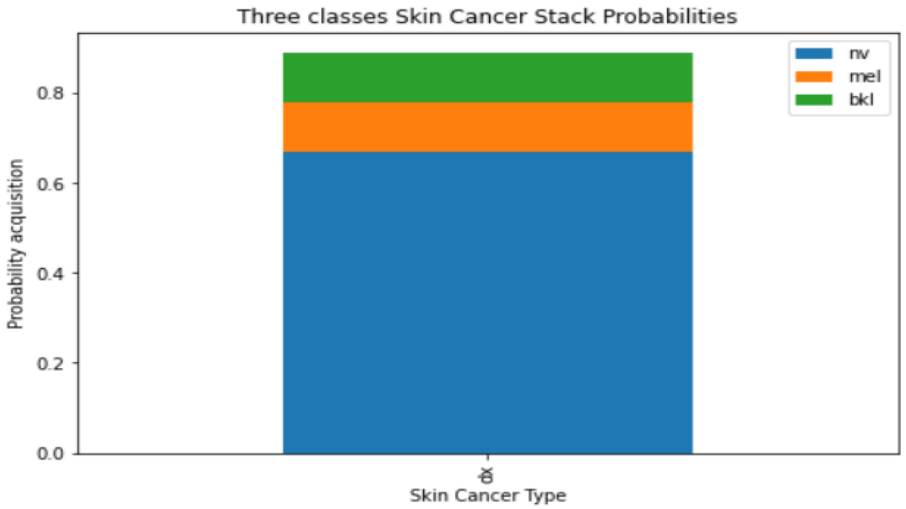


Figure 11. Skin Cancer Classes Probabilities (Authentic)

This is exactly what called **Imbalanced Data** which is a fatal trap where the data scientist may fall into. To be clear for our dataset to be served we have one class takes over about **70 percent** of the overall skin cancer dataset so if we are going to train this model, if we consider our dataset as political parties and the model as the decision to be taken, it will always be the decision of the bigger political party(**biased**).

Therefore, our model will serve the one class with more data well and forget about other two classes (small weights) as they didn't even exist. Accuracy as well will be deceiving even if it was high. There are two solutions to be made to overcome imbalanced data as possible:

- Over and Under Sampling (Random or Manual) (Not preferable)
- **Class Weighting**

Under sampling is a technique which deletes examples from the majority class which may leads to lose information from the model, on the other hand Over sampling is the opposite which duplicates the minority class images.

"Random over-sampling, a random set of copies of minority class examples is added to the data. This may increase the likelihood of overfitting, especially for higher over-sampling rates. Moreover, it may decrease the classifier performance and increase the computational effort" [11].

For class weighting we make the model pays more attention to minority class by increasing its weights by some ratio not explicitly identified but there some ways to deal with these class weighs, in our model we used this form:

$$class\ weight = \frac{1}{examples_{class}} * \frac{examples_{total}}{number\ of\ class} - const \qquad (1)$$

This const in equation is not a known value as well by we will get it by tuning to get its best value, then the question is how to know which value is? , accuracy will not be our key property as usual to determine the best model. To be clear we used ResNet101 architecture with some hyperparameters as a start to get best value for the const above first, then tuning the hyperparameters step as usual to get best model.

Using a little trick here to determine the const, is to get the accuracy of number of samples from validation test for each class. Without class weighting we got high accuracy for nv class and low accuracy for other two classes, then adding class weights to the TensorFlow model to fit and tune the const in the equation 1 to get **accuracy for each class** close as possible.

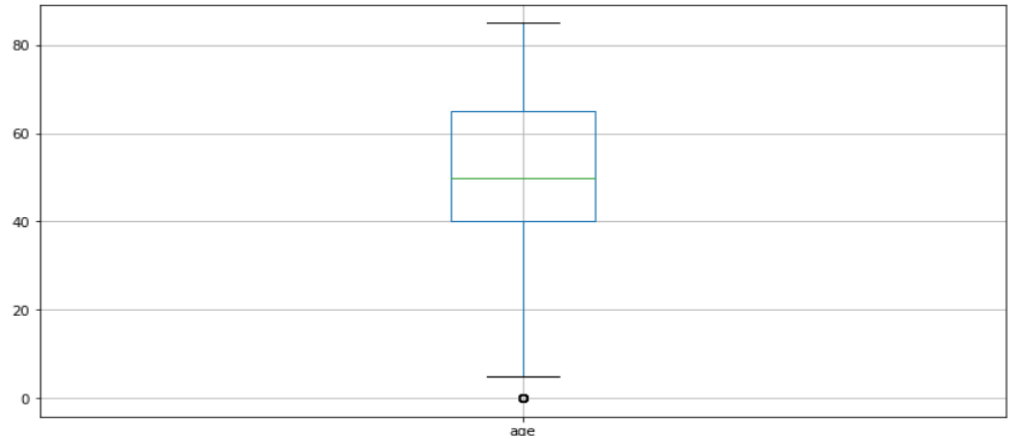


Figure 12. Boxplot for Age Feature (Authentic)

Figure 12 shows age distribution for our dataset came from metadata came with images, as predicted 1st to 3rd quartile lies from 40 to 65 and the mean is 51 years.

Image width to height ratio was 1.3333 (600,450) px originally, at first, we ignored that ratio and move with (160,160) px instead, but it might affect model learning somehow as skin cancer diagnosis is very sensitive to lesion diameter and dimensions. Therefore, we kept that ratio in other iteration (200,150) px which leads to higher rates of learning and slightly better accuracy with less epochs.

After trying different ideas for the model, we got **80 percent** overall validation accuracy, and according to [12], Skin cancer is one of the most **frequently misdiagnosed** types of cancers. Around **25%** of skin cancer cases are misdiagnosed, hence there's a limitation here.

4.2.3 Colorectal Cancer Model

Huge thank to [7] for the dataset here, which is very intuitive and balanced as it contains 5000 images 625 for class which means eight classes. 'Adipose Tissue', 'Complex Stroma', 'Debris and Mucus', 'Empty', 'Immune Cell Conglomerates', 'Mucosal Glands', 'Simple Stroma', 'Tumor Epithelium'.

Figure 13 indicates eight random images with their label taking from the training dataset which is going to be trained for our model.

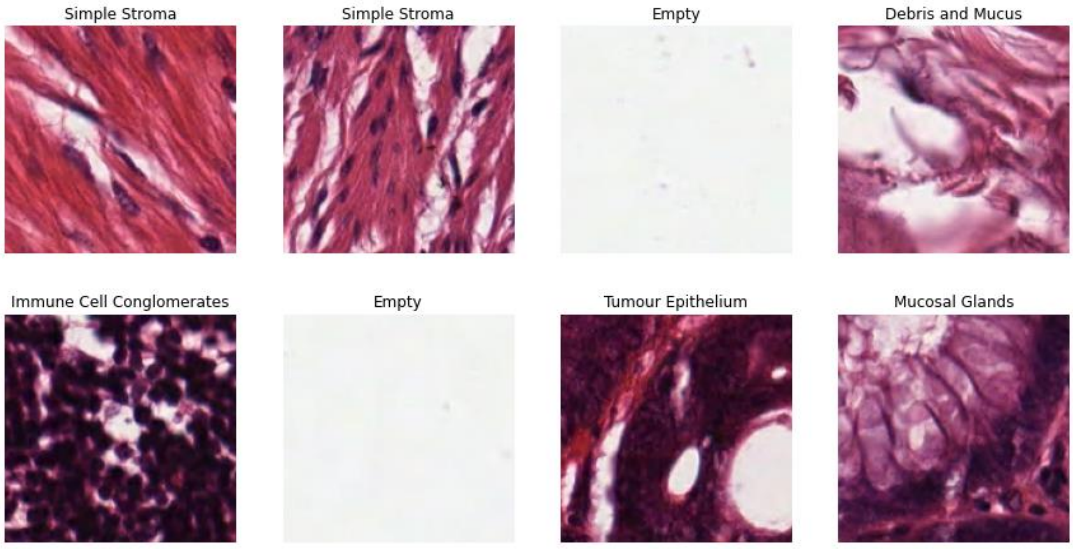


Figure 13. Eight Images for Colorectal Cancer

Using ResNet101 Model with categorical cross entropy for loss, and apply augmentation with flipping and rotations as well, we got an overall training accuracy about **98 percent,** and precision **98.5** and validation accuracy **89 percent** and precision **88.6** percent at first sight it might be indicated as overfitting with high bias. Nevertheless, we iterates different times not any epoch the validation accuracy exceeds **89 percent** even if training accuracy is low. So, it might be an indicating of the limitation of the model to be classified as better than this time.

5. DEPLOYMENT

The three final models created one for breast cancer, skin cancer, and colorectal cancer will be converted into an API using **Flask** [13] for easily use with Flutter framework, so we can use function in model to predict the cancer type with just an image uploaded by camera or from gallery and request sent to the API which get the results as a JSON file back to the mobile application for user to interact with.

5.1 Flask API

We make one API for the three different models but with three different routes to handle them in Flutter framework with POST method to upload the image to the server and get the results (predicted class, and confidence).

After Deploying the API with 0.0.0.0 host for external use and testing the API using **Postman**, we made the API ready to accept image and return results , for the sake of testing we uploaded a **nv** skin cancer type.

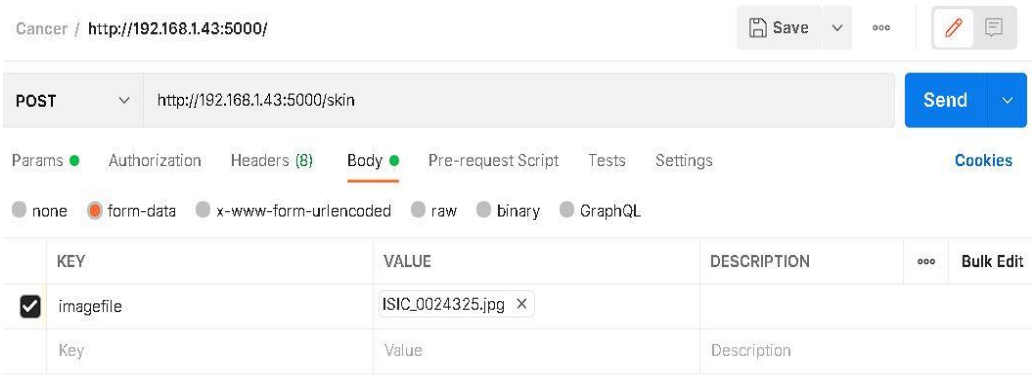


Figure 14. Postman API Testing with image uploading

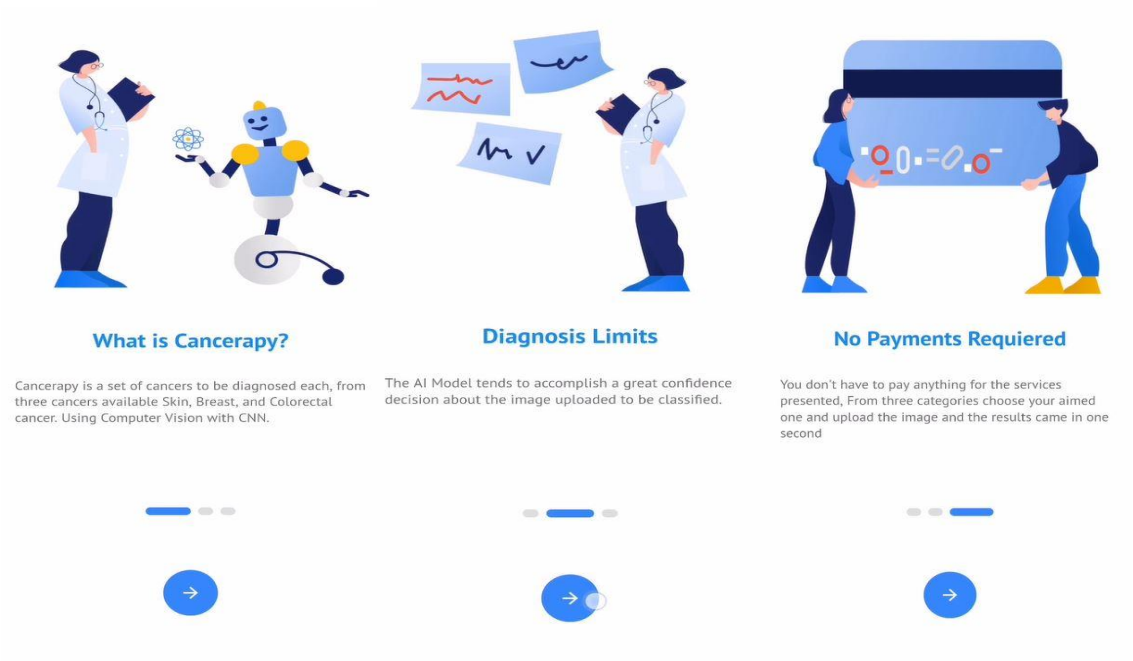
Here's the response for the image uploaded, we got **Melanocytic nevi (nv)** which exactly the same type as image uploaded for testing with 99.52 percent confidence.

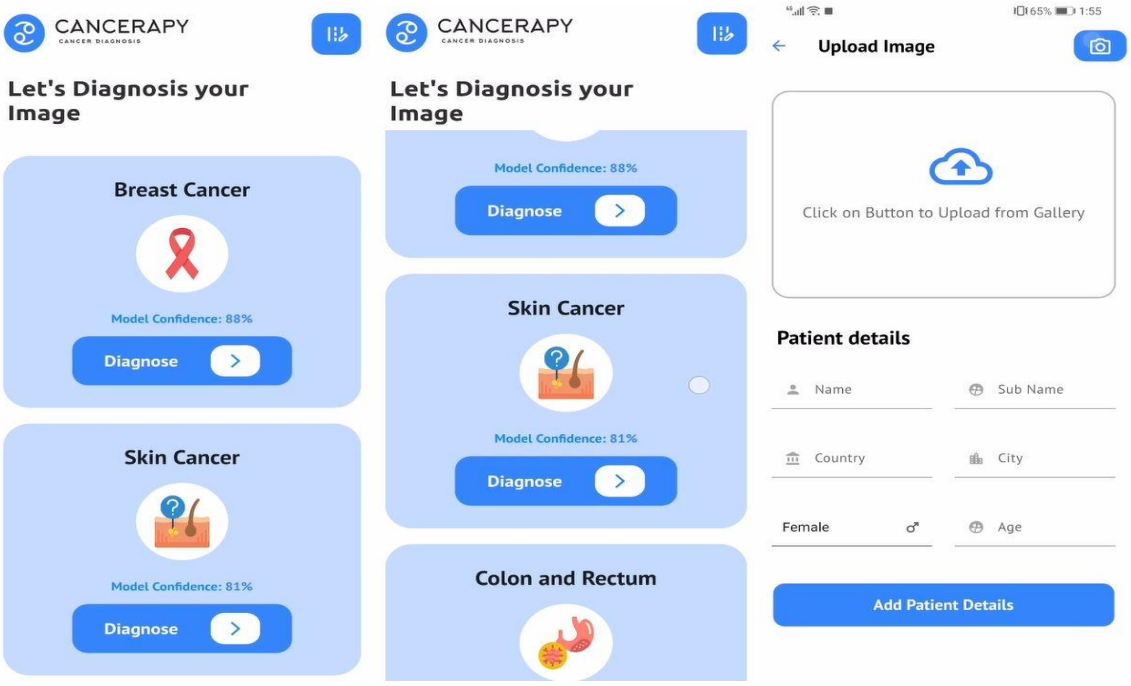
Figure 15. Postman response for image testing

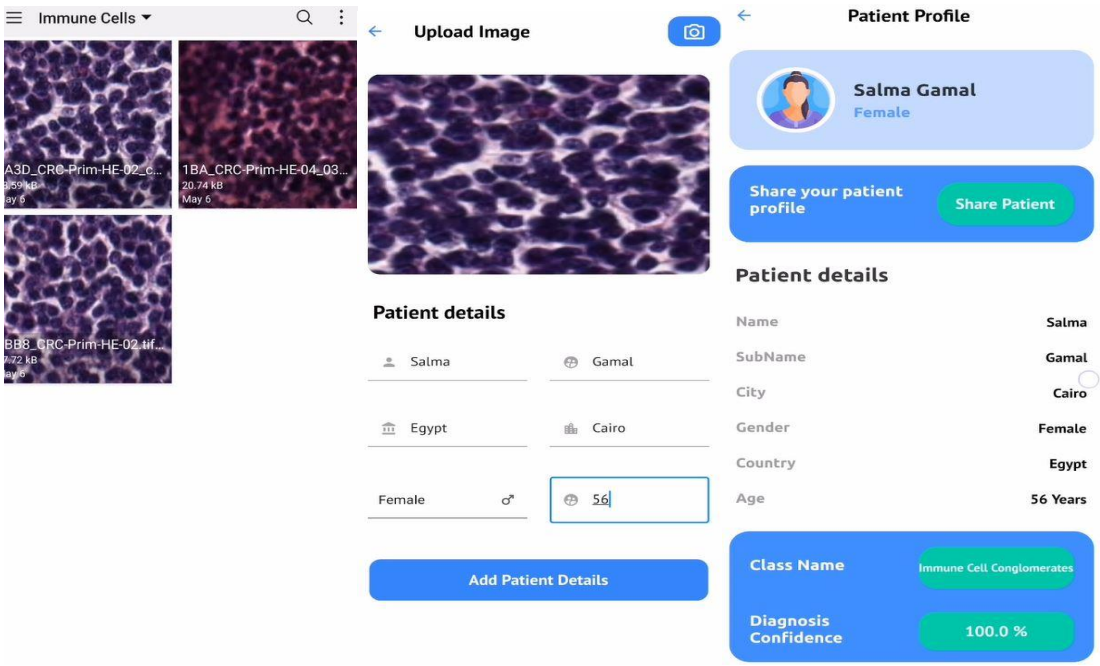
5.2 Flutter Application

Making a quite good UI with reasonable functionalities, the UI consists of about 7 Screens and there's a screen for handling the idea of uploading an image from Gallery and doing Post request for the API created, here's some screenshots for the Application created.

And for Immune cell type in colorectal cancer image uploaded, we got the response as shown in the figures below as Immune cell as well with confidence.







6. FURTHER WORK

There are still much more to do for this project to be mature:

- We must try other architectures other than ResNet101 and MobilNetv2 and comparing between them by computation time and accuracy.
- There's other trick for solving imbalanced data may work will, it's like oversampling without really duplicating images exactly, we can augment images until they catch the required number of images to solve imbalanced data.
- Increase the number of classifiers (Cancers) in the application and improves the UI of the mobile application.
- For later versions, we can take the uploaded image and classify it and added to the database with other trained images to improves the number of images to be trained. This might lead to mislabeled images but overall, it's going better.

Works Cited

[1] W. H. Organization, "World Health Organization," 3 February 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer.

[2] A. H. Abdelaziz, "Breast Cancer Awareness among Egyptian Women," *Clinical Oncology Department, Faculty of Medicine, Ain Shams University,* vol. 17, no. 1-8, 2021.

[3] J. M. A. G. S. L. A. N. a. A. J. C. E. DeSantis, "Breast cancer statistics, 2017, racial disparity in mortality by state," *Cancer Journal for Clinicians,* vol. 67, no. 6, 2017.

[4] A. C. Society, "Colorectal Cancer Facts," United States, 2022.

[5] INbreast_Database, "Medical Research," [Online]. Available: http://medicalresearch.inescporto.pt/breastresearch/index.php/Get_INbreast_Database.

[6] M. U. o. V. HAM10000, "Harvard Dataverse," [Online]. Available: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T.

[7] J. N. Kather, "Collection of textures in colorectal cancer histology," Zenodo, 26 5 2016. [Online]. Available: doi: 10.5281/zenodo.53169.

[8] "Mayo Clinic for dense breast," [Online]. Available: https://www.mayoclinic.org/tests-procedures/mammogram/in-depth/dense-breast-tissue/art-20123968.

[9] L. CD, "National performance benchmarks for modern screening digital mammography," *Breast Cancer Surveillance Consortium.,* 2017.

[10] H. M. Rolfe, "Accuracy in skin cancer diagnosis: a retrospective study of an Australian public hospital dermatology department," *Australasian Journal of Dermatology,* 2012.

[11] L. T. R. R. Paula Branco, "A Survey of Predictive Modelling under Imbalanced Distributions," *Cornell University ,* 2015.

[12] "Miller and Zois, Skin Cancer Misdiagnos," [Online]. Available: https://www.millerandzois.com/maryland-skin-cancer-misdiagnosis.html.

[13] Flask, "PalletsProjects," [Online]. Available: https://flask.palletsprojects.com/en/2.1.x/.

[14] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Sebastopol: O'Reilly Media, Inc, 2019.

[15] "TensorFlow," [Online]. Available: https://www.tensorflow.org/lite/convert.