# 

# 

# **ASSIGNMENT 2**

# **Machine Learning COSC-2753**

# 

# **Huynh Van Anh – s3836320**

# **Nguyen Vinh Quang - s3817788**

# **Nguyen The Minh - s3878434**

# **Chang Minh Cheng – s3824318**

# 

# 

# **Introduction**

Nearly 10 million people will die from cancer in 2020, making it the leading cause of death in the world.Breast, lung, colon, and prostate cancers are the most common. Tobacco use, obesity, alcohol consumption, low fruit and vegetable intake, and inactivity all contribute to a third of cancer deaths. Human papillomavirus (HPV) and hepatitis are two of the most common cancer-causing infections in low- and middle-income countries. Early detection and effective treatment are key to curing many types of cancer.

Precancerous lesions and malignant tumours are the result of a multi-stage process in which normal cells transform into cancerous ones. The interaction between a person's genetics and three types of external agents results in these modifications including ultraviolet and ionising radiation, chemical carcinogens such as asbestos, alcohol, aflatoxin, and arsenic, as well as biological carcinogens such as infections from certain viruses, bacteria or parasites.

Therefore, in task 1 our team will construct a model to determine whether a certain cell image indicates a malignant cell or not by classifying photos in this manner (isCancerous). Meanwhile, for task 2 we will build another network for classifying images according to cell-type, such as fibroblast, inflammatory, epithelial or others. For completing the tasks, we have two dataset which are main and extra, and the main data consists of 9896 images, the extra data have 10384 images(all of these images are in the size of 27x27x3). In addition, two important columns that we need to take into account are the isCancerous and the cell type column. The isCancerous column represents if the cell is cancerous or not, and the cell type column presents the label of the image.

# **Explore Data Analysis**

## Dataset

Two file of dataset:

**data\_labels\_mainData.csv** : 9896 rows & 7 columns, not contain null data

**data\_labels\_extraData.csv** : 10384 rows & 4 columns, not contain null data

**Patch\_images: png**

## Data distributed

**isCancerous**: In the main data, non-cancerous cells are 0 and malignant cells are 1. 58.8% for 0 and 41.28% for 1. According to the research, imbalanced data sets have unbalanced class proportions. Majority classes are those that make up the bulk of the data collection. Minority courses are those with a smaller proportion. The dataset is unbalanced when the minority data ratio ranges from 1% to 40%. Moreover, the degree of unbalance is mild (20-40%), moderate (1-20%), and excessive (1%) So, the difference between non-cancerous and cancerous is 17%, indicating that the primary dataset is not skewed. Therefore, the extra data is extremely imbalanced due to the percentage of cancer being nearly one third of the non cancerous, and the combined dataset also faces the same problem as the ratio between them is 0.5.

**cellType**: This graph depicts epithelial cells causing cancer. The rest of the cells are benign. Having large balance level differences leads to erroneous classifier training and prediction. Even while huge data sets have the same class ratio (imbalance level), the actual instance levels tend to climb. Minor classes are inadequately represented. We overtrain our strongest courses and undertrain our weakest. As a result, classifiers become less reliable.

This graph shows how many of each cell type there are in the primary dataset. Not cancerous. Less inflammatory cells are found in the epithelium. Undersampling and oversampling can be employed here. Common data-driven class imbalance fixes Undersampling is used to train a new class (i.e. the remainder of the data will not be used during training). Minority classes are oversampled. Initially, random sampling was used to explain class distributions.

**Scatter distributed**: Because of the dispersion in 2D, we were unable to make a clear distinction between the iscanerous 0 and 1 in the main, extra, and combined datasets. This was the case for all three of the datasets. Once the 3D graph of each file has been plotted, the visualisation is substantially improved when the cancerous dataset is plotted on top of the iscancerous (0) dataset. On the other hand, when the cancerous dataset is plotted on top of the iscancerous (0) dataset. The cell type scatter distribution in 2D is quite comparable to the study of the iscancerous 0 and 1, but in the 3D plot, the incidence of 4 cell types is somewhat similar, with the exception of cell 2. The majority of the cells, with the exception of cell 2, lean to the negative side.

## Colour distributed

We can observe that the three colours in this cell image are less mixed, indicating that the picture contains primarily pink rather than a single dominant colour. The blue curve travels a bit higher in the value range than the other colours: we observe some image having the white background.

### Edge detection

Here, we can see that the image's quality may be estimated. But to be sure that it will enrich our own model, we need to apply our procedure (HOG descriptors) on the generated image. Using the previous graphs, we can see that HOG descriptors of the edge-detected image are better able to identify the cell type than the original image. The classification task can now be launched to see if it has any effect on the algorithm itself.

# **Preprocessing**

## PCA

PCA is a technique for transforming a dataset's columns into new features known as Principal Components. As a result, a significant amount of data from the entire dataset can be condensed into fewer feature columns. This allows for dimensionality reduction and the visualisation of the separation of classes or clusters if any are present.

Using it, the dataset's feature redundancy can be eliminated. Also, it seeks to gather vital information that explains the high variance, which results in the best accuracy. It simplifies the process of working with data visualisations. By doing so, the model's complexity is reduced, but its computing efficiency is improved.

## Image Flattening

Flattening is a technique for converting multi-dimensional arrays into a 1-dimensional array. It is commonly used in Deep Learning while feeding the 1-D array information to the classification model.

One of the most important reasons why we flatten the Image Array before processing is because multidimensional arrays require more memory than 1-dimensional arrays. Flattening helps to reduce the amount of memory and time needed to train the model in most circumstances because we are dealing with a big number of photos.

The memory utilised by the multi-dimensional image array and the flattened array is nearly identical once the algorithm has been run in its entirety. So, why bother doing the flattening if it has no effect at all? As the number of photos in a dataset grows, so does the amount of memory that can be saved by eliminating them.

## Data Augmentation SMOTE

Unbalance can be corrected with oversampling using SMOTE (synthetic minority oversampling technique). Randomly expanding minority class samples by copying them is the goal of this method. To create new minority cases, SMOTE blends together existing minority cases. Linear interpolation is used to create the virtual training records for the minority class. Each example in the minority class has a random selection of one or more of its k-nearest neighbours to create these synthetic training records. Several categorization models can be applied to the processed data after the oversampling procedure.

The following is how the SMOTE algorithm operates:Draw a random sample from the minority group. K nearest neighbours will be identified from this sample.Once you've chosen a neighbour, you'll need to figure out the direction of travel between the current data point and that neighbour. The vector is multiplied by a number between 0 and 1, and the result is the new vector. For the synthetic data point, you add this to the current data point. This is essentially the same as moving a data point in the direction of a neighbouring data point. Your synthetic data point will not be a perfect clone of a known data point, but it will also not be too distinct from known observations in your minority group.

In task 1 we need to classify cell images to figure out if the patient is cancerous or not. Moreover, we do not apply the data augmentation to increase the number of images in a dataset because the quantity of the data is enough (20280 images for training and validation). Therefore, smote is applied due to the imbalance problem between the cancerous and not cancerous class.

## Test-spit and how to evaluation

Hold-out cross validation is used to divide the data set into a training and testing set. It is necessary to utilise both a training set and a testing set to determine the model's correctness. Model generalisation is said to be good if training and accuracy are almost same. For our model, 80 percent of the data is used for training, and the remaining 20 percent is used for validation purposes.

The reason that we apply hold-out cross validation is when you're working with a large dataset, short on time, or just starting to create an initial model for your data science project, the hold-out method is an excellent choice. Keep in mind that the other cross-validation requires more processing power and time to run than the holdout approach because it employs numerous train-test splits.

Our model in task 1 is used for classifying images according to whether a given cell image represents a cancerous cell or not (isCancerous). The precision is the positive cases accurately detected from all the projected positive cases is implied by this metric, and when the costs of false positives are considerable, it is useful. However, we do not want to miss any case of real cancer so it is not possible to apply the precision. Recall is also not possible at all, imagine that we predict a non cancerous cell to become cancerous, it causes a serious mental damage if we predict a cell for a patient, and we must minimise this problem. Another matrix is accuracy, assuming that there are only 30 people in the sample above who are in fact cancerous. How would you feel if our model revealed that 25 of the individuals had cancer so in this situation, the accuracy is 90 percent, which is high enough to be deemed 'correct'. 5 people truly had cancer, although the model predicted that they didn't. This is clearly an expensive trade off. False Negatives should be minimised in our model. Therefore, the F1-score is used in these situations where precision and recall cannot be reliably determined, the harmonic mean of precision and recall can be used instead. Furthermore, F1-score is used when the False Negatives and False Positives, and this is the matrix we apply for our model.

# **Modelling**

## Task 1

***Support Vector Machine***

Support Vector Machine is a method for supervised machine learning that can be used to solve classification and regression problems (Vegi Shanmukh 2021). But it is often used to sort things into groups. In this algorithm, each piece of data is shown as a point in n-dimensional space, where n represents the size of features (Vegi Shanmukh 2021). The value of each feature is the value of a certain coordinate. Then, we classify the data by finding the hyperplane that best separates the two classes (Vegi Shanmukh 2021).

Why does SVM get picked for task 1? As the EDA section noticed, our isCancerous image class is spread out in different places, and SVM performs very well with a considerable margin of separation (SVM | Support Vector Machine Algorithm in Machine Learning 2017).

According to Vegi Shanmukh 2022, some of the most important SVM parameters are :

* **Gamma**: shows how far a single training example can change the values, which can lead to skewed results.
* **C**: Keeps the cost of mistakes in check
* **Kernel**: The kernel is a group of arithmetic operations that SVM algorithms use. Kernels come in three different kinds: linear, RBF and polynomial.‌

Our strategy for applying SVM is first we need to convert the images dataset to 2D because SVM takes 2D input to train. Then split data into training and test sets. Applying the technique of image flattening, normalisation (reduce the training time) and Gridsearch to receive the best performance metrics .

***Convolutional Neural Network***

Firstly, what is CNN? CNN stands for convolutional neural network is a class of deep neural network, which is commonly used for analysing images. It uses a technique called Convolution to analyse the image. Convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.(Vidhya, 2022)Convolutional neural networks are composed of multiple layers of artificial neurons that are like biological neurons in taking in multiple input and output an activation value. For example, if our eyes see a circle, it gives input to the brain how that object looks and our brain processes that image and recognizes it is round because it looks like a circle. It is the same for CNN, by creating a panel with basic features like round, it will apply on the image to see if it has this feature or not.

Usually, the first layer of CNN usually analyses the basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As more layers are added, the more complex we can analyse.

For our CNN model, we have 9 layers in total with 4 2-dimensionals Convolutional layers, 2 2-dimensionals maxpool layers, and 3 Dense layers. Other than that, our model also applies layers and implementations such as Batch Normalisation, Flattening, Dropout. Batch Normalisation is for transforming data, Flattening is for changing the data shape for training and finally, Dropout is to drop random data so that the model will not be over-fitting and taking in wrong data. (Doshi, 2021) (Brownlee, 2016)

The model will take in a RGB image which has the height and width equal to 27x27 and categorise based on either the cell type or does it have cancer or not.

## Task 2

**Analysing**

The second task of the project is aiming to construct a model that distinguishes images into 4 types of categories. Unlike task one, more than half of the data have not been labelled yet so we are unable to perform the supervised learning technique in this task. Based on our research, we find out that semi-supervised learning is the key to solving the problem. In brief, semi-supervised learning (SSL) is a machine learning technique that employs a little amount of labelled data and a large quantity of unlabeled data to train a forecasting model (AltexSoft).

There are many methods, but we decided to use self-training, because it is the simplest example of semi-supervised learning, and we see that its workflow is suitable for our case which will be explained later in the report. Self-training is the procedure by which any supervised algorithm for classification or regression can be modified to operate semi-supervised, utilising both labelled and unlabeled data.(AltexSoft)

**Dataflow of self-learning**

Similar to supervised learning, train the model with a modest amount of labelled training data until it produces satisfactory results. The best model is then used with the unlabeled training dataset to predict the outputs, which are pseudo labels because they may not be entirely precise. Then set a confident value in order to collect the most confident predictions. If any of the pseudo-labels are above this confidence threshold, they are added to the labelled dataset and create a new, combined input. After that, we will use these updated datasets to train an improved model. This process may complete multiple iterations with more and more pseudo-labels being added. Assuming the data is suitable for the procedure, the model's performance will continue to improve with each repetition. (AltexSoft)

**Implement**

In our project, we will measure two CNN models with different architecture before starting the self-learning process. Finding which one has a better performance based on the validation loss and its accuracy. The reason we do this step is to find out the appropriate number of CNN layers which will treat overfitting in our model (Sagar). When we have successfully chosen the right model, it will be put into the self-training process. We decide to make 10 iterations or when the unlabeled dataset is empty, the process will stop. There are two reasons we choose only to run at most 10 iterations. The first one is based on AltexSoft, a technology consulting company, 10 is a standard amount in self-learning (AltexSoft). The second reason is we have tested the process and on the tenth iteration, the accuracy of the model has already been very high (85%) and the unlabeled dataset only reminds around 200 data while the original one is more than 10,000. The last thing is the confidence threshold, we decide to accept the prediction that is equal to or higher than 75%.

# **Hyperparameter Tuning**

## Grid Search

It is a librarian feature in the model selection package of sklearn. It assists to loop through preset hyperparameters and fit your estimation model to your training set. So, throughout the finish, you can choose the best parameters from those that are listed.

**SVM**

**Kernel**: For a linear kernel, the following formula is used to anticipate a new input based on the dot product of the input (x) and each support vector (xi): ***f(x) = B(0) + sum(ai \* (x,xi)).*** The learning algorithm must figure out, from the training data, what the coefficients B0 and ai are for each input. With polynomial and exponential kernels, the isolation boundary can be calculated in extra dimensionality. **Regularisation**: informs the SVM enhancement how much you'd like each training example to be correctly classified. For big values of C, the improvement will use a hyperplane with a tiny proportion if it does a decent job of accurately identifying all the training points. On the other hand, if C has a very lower amount, the optimizer will look for a separating hyperplane with a larger margin, even if that hyperplane misclassified more points. The **gamma** parameter controls how far a single training example has an effect. Low values suggest "far," while high values imply "close." In other words, when gamma is limited, points far outside the feasible separation line are used to figure out where the separation line should be. When gamma is high, the points close to the plausible line are used in the estimations.

Choosing the best parameter: applying gridSearch and visualising, we recognize that our model best performs at  **kernel='rbf', gamma=0.01, c= 10.**

**CNN**

Since we are dealing with a lot of data at more than 20,000 images, there is no need to make a complete model as soon as possible so after considering all of our options, we decided to use Adam as our optimizer. Adam is all about momentum and finding all the little detail which is perfect for our model.(Doshi, 2019) The Adam optimizer is not good if you do not have a lot of data and if you do, Adagrad optimizer might be better but the thing that Adam is different from Adagrad is that it looks for other details and searches carefully. We have enough data that we can use Adagrad and Adagrad will be perfect if we are analysing pictures such as a car or a bird, but cell image is different. We are looking for every single detail since that little detail can separate one cell type from the other so Adam, which not only works well when we have a lot of data but also can find small details that can tell the cell apart.

For tuning the parameter, since we have a large dataset, I would prefer using a small learning rate with a decently high epsilon value so that the model will not be overfitting and working just fine. I do not want the learning rate to be high because a high learning rate can change the model too much and with a large dataset, it will become overfitting. As for epsilon, we will be looking for a decently high epsilon value so that we will be able to achieve better results. For that reason, we will be tuning the parameter for learning rate from 1.0e-5 to 1.0e-4, and for epsilon, we will be taking the result from 1.0e-7 to 1.0e-2. We will be running on 5000 data which should give us a good overview on how the result will turn out and then we will consider which one will be the best for us since we will be training on 2-4 times more data than when we find the best parameter.

# **Evaluation**

## Compare two model task 1

Limitation of SVM applying Grid

SVM: The best SVM model is SVM base with SMOTE technique, giving the F1 score is 0.839 and Recall score is 0.8458. The GridSearch app has improved the Precision score but not Recall and F1 score (which are our priorities).This hyperparameter was made to handle class imbalance, which is the same as controlling the trade-off between precision and recall. The SVM also has drawbacks related to the training time, it takes a long time for a large dataset. For a larger dataset, CNN is another option that we can consider. Gridsearch also takes a very long time to execute but at the end we do not receive the expected result.

Unexpectedly, the base CNN model has the highest score out of all the CNN models. It has an average score at 0.91 which is astonishingly high. The second best performing model is the CNN model using SMOTE to balance data out but was using the same optimizer and architecture so from this, we can conclude that using SMOTE technique did not benefit in training the CNN model. The reason for this might be because SMOTE technique will be highlighting the common features and might have ended up disregarding the distinct features. As for changing the optimizer, we have 2 models that have different optimizers compared to the original one but both of them perform worse than the original model. The reason for this might be because we had a vision too far off and had chosen a learning rate that is too low and a too high epsilon value for the amount of data that we had which ended up making the model not performing as good as it could have been. Nevertheless, all of the CNN models outperform the SVC model, so we can conclude that using the CNN model in this case is better than the SVC model in predicting whenever the given cell image indicates that it has cancer or not.

SVM (Pros & Cons):

* Pros:
* Computational intensive
* Perform effectively in high dimension space
* With an appropriate kernel function, we can solve any complex problem
* Cons:
* High training time for large datasets
* Difficult to understand and interpret the final model, variable weights and individual impact
* Choosing an appropriate Kernel function is difficult

CNN (Pros & Cons):

* Pros
* Neural networks can be trained with any number of inputs and layers.
* Neural networks are flexible and can be used for both regression and classification problems.
* Neural networks work best with more data points
* Cons
* If the CNN has several layers then the training process takes a lot of time if the computer doesn’t consist of a good GPU
* Neural networks are black boxes, meaning we cannot know how much each independent variable is influencing the dependent variables.
* Neural networks depend a lot on training data. This leads to the problem of overfitting and generalisation. The mode relies more on the training data and may be tuned to the data.

## Compare models task 2

Model CNN 1: The first CNN model has 6 layers. 3 layers is for analysing the features that each picture has while the next 2 layers is for finding all the variations of all the features that the picture has and highlighting them before finally categorising them to find which cell type it is. This model focused on simplifying and did not focus on analysing all the features as the model in task 1 but focus on getting the general picture and getting all the variation.

Model CNN 2: The second CNN model has 10 layers. The first 6 layers are for analysing the features that each picture has while the next 3 layers are for finding all the variations of all the features. This model is much more focused on finding all the features and minimising them and finding all the variation and then using it to categorise them based on the cell type. This model has poorer performance compared to the first model because it is analysing the features too much and it is also focused too much on finding all the variation of the features that the common feature does not get focused as much as how the model 1 did.

**Choose best model**

We have 2 base CNN models and 2 pairs of target and output, one has been applied data augmentation (SMOTE) one is not. Therefore, we will have overall 4 models to train and test on labelled data before choosing the best model for applying self-training. After we have trained all of them with epochs 40, learning rate 0.00006 and epsilon 1.0e - 7. Surprisingly, the model 1 without SMOTE has the highest f1 score is 0.7. It seems like SMOTE is not helpful in our case and also the CNN model with denser architecture. Both of these limitations will be identified below.

Limitations of model CNN 2: In model 2, we have 10 layer which 4 more layer than model 1. More layer on the CNN model can cause overfitting which lead to the decreasing of the accuracy (3).

Limitations of CNN applying SMOTE:

SMOTE is a technique that balances data by creating synthetic data points to increase the number of observations in the minority class. (Kumar, 2021) People trusted and used SMOTE when dealing with an imbalanced dataset because it created new data based on the already existing data. Nevertheless, it still has its own disadvantages. As it was creating new data based on the existing data, it also copied bad data that has features that are not good for the model.(Jiang, Pan, Zhang and Yang, 2021) With too much bad data, the model prediction will be off and especially with data about cell images which are very similar to each other and small overall, a lot of wrong data can be copied which leads to the overall score getting reduced.

## Compare to Paper Thesis Output:

In this experiment, they detect and classify nuclei. This research presents a SC-CNN for nucleus detection. It determines the possibility that a given pixel represents the nucleus' core, with high probability values confined to the nuclei's cores. The Neighboring Ensemble classifies nuclei. Together, NEP and CNN can better forecast the future. The nuclei's class is known. The suggested method does not require segmentation to detect and classify nuclei.

The Colorectal adenocarcinomas with more than 20,000 annotated nuclei belonging to four different classes were analysed by the researchers. 7 722 epithelial, 5 712 fibroblast, 6 971 inflammatory and 2 039 other nuclei are included in the dataset. In the documentation the researcher applies about 5 algorithms which are SC-CNN, Softmax nucleus classification, K-fold cross validation, SSPP, and NEP.

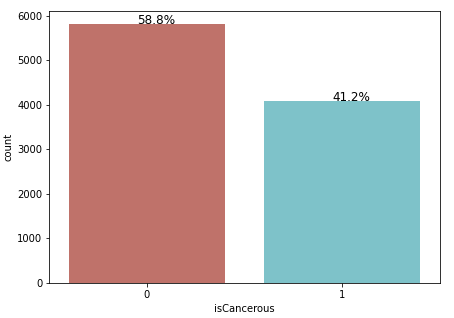
* This is the output of the detection task: The SC-CNN performs better than other models. (F1 score: 0.802)
* This is the output of the classification task: The softmax CNN + NEP have the best performance. (F1 score: 0.784)
* This is the output of the combination of detection & classification: The SC-CNN (detection) & softmax CNN + NEP(classification). (F1 score: 0.692)

After 10 iterations of self-training, the macro f1 score of our model is 0.69. Although we have applied a semi-supervised technique to attempt to get a better model, the performance of the model is unchanged. Based on the above brief, we have unsuccessfully achieved a model that is better than the softmax CNN + NEP model on classification cell type. We have come up with a judgment on the model:

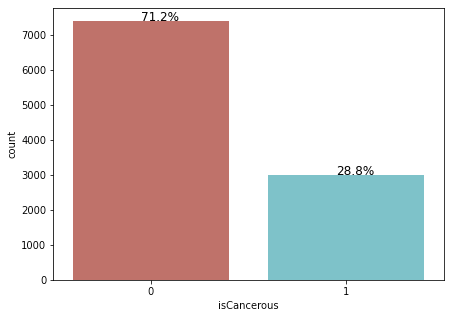
Although both datasets have almost the same amount of images, more than half of them are unlabelled so we must apply semi-supervised learning to finish the task. According to AltexSoft, the performance of the SSL model depends on the base model that is trained on the labeled dataset (AltexSoft). If the performance has already been bad at the beginning, it will lead to inferior performance. Although we have tried many methods like data augmentation (SMOTE) or tried different architecture complexity CNN models, we cannot get a better classifier. Therefore, we accept the result and try to look for new method to increase the performance like applying class weight instead of SMOTE.

# **Appendix**

**Figure 1.** Data distributed of isCancerous image class  
in main file



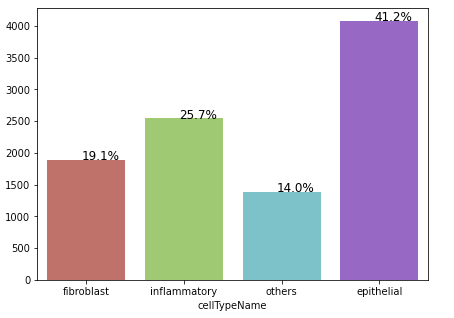
**Figure 2.** Data distributed of isCancerous image class in extra file



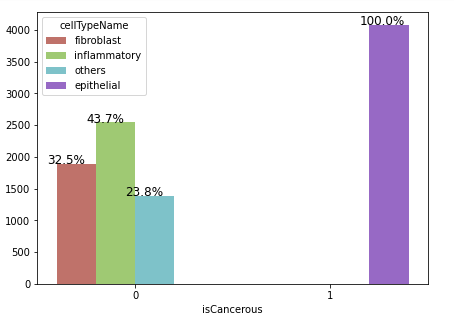
**Figure 3.** Data distributed of isCancerous image class in combination

# 

**Figure 4.** Data distributed of Cell type image class



# 



# 

**Figure 5. Some images of cells.** ****

**Figure 6. Colour distributed**

# 

# 

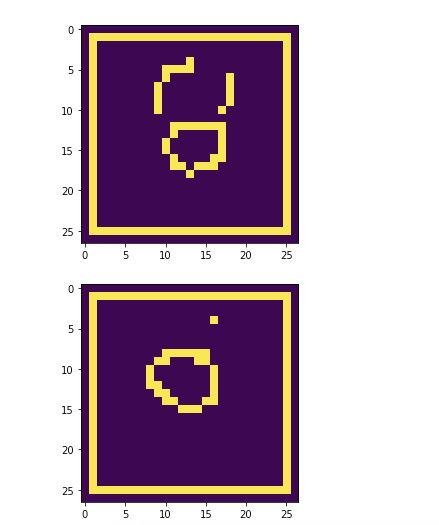
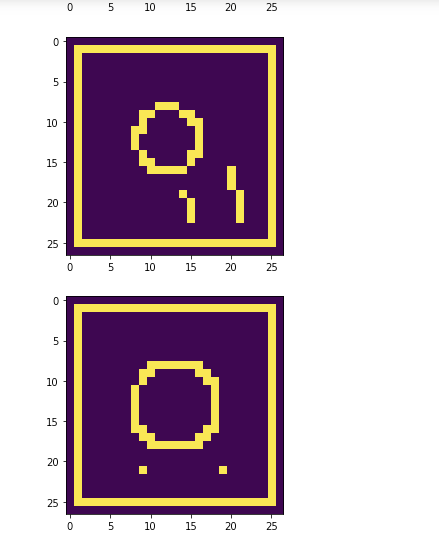
# 

# 

# 

# 

**Figure 6. Edge detection¶**



**Figure 7.** 3D distribution of isCancerous image class

| Image of class isCancerous = 0 (Negative) | Image of class isCancerous = 1 (Positive) |
| --- | --- |
|  |  |
|  |  |

**Figure 8.** 3D distribution of cellType image class

| **Image of cellType 0 & 1** | **Image of cellType 2 & 3** |
| --- | --- |
|  |  |
|  |  |

# **Reference**

Vegi Shanmukh 2021, *Image Classification Using Machine Learning-Support Vector Machine(SVM)*, Medium, Analytics Vidhya, viewed 12 May 2022, <<https://medium.com/analytics-vidhya/image-classification-using-machine-learning-support-vector-machine-svm-dc7a0ec92e01>>.

*SVM | Support Vector Machine Algorithm in Machine Learning* 2017, Analytics Vidhya, viewed 12 May 2022, <<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/> >.

Dickson, B 2021, ‘What is data augmentation?’, Bdtechtalks, 27 November, viewed 2 May 2022, <<https://bdtechtalks.com/2021/11/27/what-is-data-augmentation/?fbclid=IwAR11Hv9NGkk41Cu1JdWLcQKAEjJ16vOUER1kGV4HAlzZjSXTGkQC_jDpDKM#:~:text=This%20is%20called%20%E2%80%9Cdata%20augmentation,for%20the%20%E2%80%9Cduck%E2%80%9D%20class> >.

Gondhalekar, A 2020, ‘Data Augmentation — Is it really necessary?’, Bdtechtalks, 24 March, viewed 2 May 2022, <<https://bdtechtalks.com/2021/11/27/what-is-data-augmentation/?fbclid=IwAR11Hv9NGkk41Cu1JdWLcQKAEjJ16vOUER1kGV4HAlzZjSXTGkQC_jDpDKM#:~:text=This%20is%20called%20%E2%80%9Cdata%20augmentation,for%20the%20%E2%80%9Cduck%E2%80%9D%20class> >.

Google Developers 2021, ‘Imbalanced Data | Data Preparation and Feature Engineering for Machine Learning’, Google Developers, 1 November, viewed 3 May 2022, <<https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data?fbclid=IwAR377zM8TFK41MJWmUAaGHxkLfeu9XFDOLxH7U8P3JWjvXHThDiuIv_34cI> >.

Takimoglu, A 2021, ‘What is Data Augmentation? Techniques & Examples in 2022’, AI Multiple, 20 April, viewed 2 May 2022, <<https://research.aimultiple.com/data-augmentation/?fbclid=IwAR2G-scojyAm2g9zplF_T-vWqidNYINvlHd9Zcn9FEtq1mAfglbhVMZHWcU>>.

Hackernoon 2021, ‘What is Data Imbalance in Machine Learning?’, HackerNoon, 2 June, viewed 2 May 2022, <<https://hackernoon.com/what-is-data-imbalance-in-machine-learning-932f34n8?fbclid=IwAR18U19w93kqjhZmRbXqN5KRTjFfkVodM4RC-XvpqG27l0D96b_OhW8KCOk> >.

GeeksforGeeks 2021, ‘ML | Handling Imbalanced Data with SMOTE and Near Miss Algorithm in Python’, GeeksforGeeks, 13 December, viewed 2 May 2022, <<https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/?fbclid=IwAR3nbCrsVGzTtSbzj6OZifeMa2CY-fDR7SXad6iLxmRkzDndVr_A8bR9MMc#:~:text=SMOTE%20(synthetic%20minority%20oversampling%20technique)%20is%20one%20of%20the%20most,instances%20between%20existing%20minority%20instances> >.

Prabhakaran, S 2019, ‘Principal Component Analysis (PCA) – Better Explained’, Machine Learning Plus, 23 March, viewed 5 May 2022, <<https://www.machinelearningplus.com/machine-learning/principal-components-analysis-pca-better-explained/> >.

Pulagam, S 2020, ‘All you need to know about PCA technique in Machine Learning’, Towards Data Science, 16 Jun, viewed 5 May 2022, <<https://towardsdatascience.com/all-you-need-to-know-about-pca-technique-in-machine-learning-443b0c2be9a1> >.

Www3.tuhh.de . 2022. Edge detection — Data Quality Explored, viewed 12 May 2022<<https://www3.tuhh.de/sts/hoou/data-quality-explored/2-2-2-edge.html> >

Www3.tuhh.de. 2022. Edge detection — Data Quality Explored, viewed <<https://www3.tuhh.de/sts/hoou/data-quality-explored/2-2-2-edge.html> >

Kumari, S 2021, ‘Edge detection’, Mechmotive, 20 November, viewed 5 May 2022, <<https://mechomotive.com/edge-detection/> >.

WHO 2022, ‘Cancer’, WHO, 3 February, viewed 5 May 2022, <<https://www.who.int/news-room/fact-sheets/detail/cancer> >.

Korstanje, J 2021, ‘SMOTE’, Towards Data Science, 30 August, viewed 5 May 2022, <SMOTE | Towards Data Science>.

Analytics Vidhya. 2022. *CNN for Deep Learning | Convolutional Neural Networks*, viewed 30 April 2022 <<https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/#:~:text=In%20deep%20learning%2C%20a%20convolutional,a%20special%20technique%20called%20Convolution> >

Doshi, K., 2021. *Batch Norm Explained Visually — How it works, and why neural networks need it*, Medium, viewed 11 May 2022, <<https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739> >

Brownlee, J., 2016. *Dropout Regularization in Deep Learning Models With Keras*, Machine Learning Mastery, viewed 12 May 2022, <<https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/> >

Doshi, S., 2019. *Various Optimization Algorithms For Training Neural Network*, Medium, viewed 11 May 2022, <<https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6#:~:text=Adam%20is%20the%20best%20optimizers,descent%20is%20the%20best%20option>. >

AltexSoft 2022, *Semi-Supervised Learning, explained with examples*, AltexSoft, viewed 10 June 2022, <[Semi-Supervised Learning, Explained | AltexSoft](https://www.altexsoft.com/blog/semi-supervised-learning/)>

Sagar, A 2019, *5 Techniques to Prevent Overfitting in Neural Networks.* KDnuggets, viewed 10 June 2022 <[5 Techniques to Prevent Overfitting in Neural Networks - KDnuggets](https://www.kdnuggets.com/2019/12/5-techniques-prevent-overfitting-neural-networks.html)>

Ruizendaal, R 2017, *Deep Learning #3: More on CNNs & Handling Overfitting,* Medium, viewed 10 June 2022 <[Deep Learning #3: More on CNNs & Handling Overfitting | by Rutger Ruizendaal | Towards Data Science](https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d)>

Kumar, S., 2021. *Stop Using SMOTE to Treat Class Imbalance. Take This Intuitive Approach Instead*, Medium, viewed 14 May 2022 <<https://towardsdatascience.com/stop-using-smote-to-treat-class-imbalance-take-this-intuitive-approach-instead-9cb822b8dc45>>

Jiang, Z., Pan, T., Zhang, C. and Yang, J., 2021. A New Oversampling Method Based on the Classification Contribution Degree. *Symmetry*, 13(2), p.194.