INTERNATIONAL COLLEGE OF BUSINESS AND TECHNOLOGY

BSC (Hons) Software Engineering

MODERATED BY CARDIFF METROPOLITAN UNIVERSITY

Module Title -	**Computational Intelligence**
Module code -	**CIS6005**
Name -	**Mushad Aadil**
Student ID -	**KD/BSCSD/07/01**
Branch –	**ICBT KANDY CAMPUS**

SUBMITTED TO: Cardiff Metropolitan University Moodle

SUBMITTED ON: 02nd June 2019

<table>
<tr><th colspan="7">Student Details (Student should fill the content)</th></tr>
<tr><td>Name</td><td colspan="6">Mushad Aadil</td></tr>
<tr><td>Student ID</td><td colspan="6">KD/BSCSD/07/01</td></tr>
<tr><th colspan="7">Scheduled unit details</th></tr>
<tr><td>Unit code</td><td colspan="6">CIS6005</td></tr>
<tr><td>Unit title</td><td colspan="6">Computational Intelligence</td></tr>
<tr><td rowspan="2">Unit enrolment details</td><td>Year</td><td colspan="5">3</td></tr>
<tr><td>Study Period</td><td colspan="5">2019</td></tr>
<tr><td>Lecturer</td><td colspan="6">Gihan Herath</td></tr>
<tr><td>Mode of delivery</td><td colspan="6">Full Time</td></tr>
<tr><th colspan="7">Assignment Details</th></tr>
<tr><td>Nature of Assessment</td><td colspan="6">Coursework (Project & Report)</td></tr>
<tr><td>Final Grade Weighting</td><td colspan="3">Project: 25%</td><td colspan="3">Report: 25%</td></tr>
<tr><td>Topic of the Case Study</td><td colspan="6">AI based Software Application – CW 1</td></tr>
<tr><td>Learning Outcomes covered</td><td colspan="6">LO1, LO2</td></tr>
<tr><td>Word count</td><td colspan="6">3000</td></tr>
<tr><td>Due date/ Time</td><td colspan="6">02nd Jun 2019 / 12 A.M</td></tr>
<tr><td>Extension granted?</td><td>Yes</td><td>No</td><td colspan="2">Extension Date</td><td colspan="2"></td></tr>
<tr><td>Is this a resubmission?</td><td>Yes</td><td>No</td><td colspan="2">Resubmission Date</td><td colspan="2"></td></tr>
<tr><th colspan="7">Declaration</th></tr>
<tr><td colspan="7">I certify that the attached material is my original work. No other person’s work or ideas have been used without acknowledgement. Except where I have clearly stated that I have used some of this material elsewhere, I have not presented it for examination / assessment in any other course or unit at this or any other institution</td></tr>
<tr><td>Name/Signature</td><td colspan="3"></td><td>Date</td><td colspan="2">02/06/19</td></tr>
<tr><th colspan="7">Submission</th></tr>
<tr><td>Return to:</td><td colspan="6"></td></tr>
<tr><th colspan="7">Result</th></tr>
<tr><td>Marks by 1st Assessor</td><td></td><td colspan="2">Name & Signature of the 1st Assessor</td><td></td><td colspan="2" rowspan="2">Agreed Mark</td></tr>
<tr><td>Marks by 2nd Assessor</td><td></td><td colspan="2">Name & Signature of the 2nd Assessor</td><td></td></tr>
<tr><td>Comments on the Agreed mark</td><td colspan="6"></td></tr>
</table>

ASSIGNMENT FEEDBACK SHEET

<table>
<tr><td colspan="2">STUDENT NAME:</td><td>STUDENT NUMBER:</td></tr>
<tr><td colspan="2">Module Number & Title:</td><td>Semester: II</td></tr>
<tr><td colspan="3">Assignment Type & Title:</td></tr>
<tr><td colspan="3">For student use: Critical feedback on the individual progression towards achieving the assignment outcomes</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3"><u>For the Assessors’ feedback</u>
Indicate the Task number strength and Weakness and the marks for each task</td></tr>
<tr><td>Task No/ Question No</td><td>Strengths (1st Assessor)</td><td>Strengths (2nd Assessor)</td></tr>
</table>

Task No / Question No	**Weaknesses (1st Assessor)**	**Weaknesses (2nd Assessor)**

Areas for future improvement	
Comments by 1st Assessor	**Comments by 2nd Assessor**

Marks				
Task /Question No	**Marks by 1st Assessor**	**Marks by 2nd Assessor**	**Marks by IV (if any)**	**IV comments (If Any)**
Total Marks				

Name and the Signature of the 1st Assessor		**Date:**	
Name and the Signature of the 2nd Assessor:		**Date:**	
Name and the Signature of the IV: (if any)		**Date:**	

ACKNOWLEDGEMENT

I acknowledge Mr. Gihan Herath – the assigned lecturer who has taught me this module. And all the other staff at the International College for Business and Technology Kandy.

EXECUTIVE SUMMARY

- In this report I have written the literature review of the Artificial Intelligence Software I have developed with the help of TensorFlow library and the Keras API in Google Colabatory. I designed and developed an AI that can detect up to five different types of diseases. They include two types of skin cancers, a cut and a bruise. As the model is designed for medical diagnosis it can be easily trained to detect other diseases.
- I have explained the system architecture and talked about a different model developed by the University of Stanford.
- The AI technique used and the theory behind that.

CONTENTS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LITERATURE REVIEW & SYSTEM ARCHITECTURE

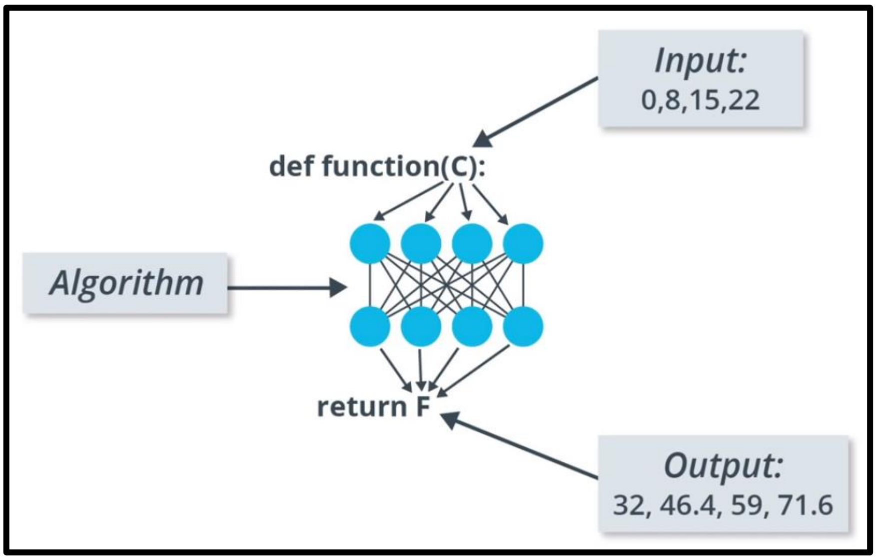
I developed an Artificial Neural Network based classification application that can diagnose different types of medical conditions. The application can detect between two types of cancer (Benign and Melanoma), a cut and a bruise. Skin caner is the second leading cause of cancer deaths in America. With more that 5.4 million people reportedly developing symptoms every year. With a single skin lesion the model will be able to predict the cancer with human level accuracy.

Software

- Keras – Keras is a high-level Application Programming Interface (API) which is extremely user friendly
- TensorFlow – TensorFlow is an opensource library for programming. Specifically, machine learning. It contains various premade AI models which can be directly imported into your project.
- Google Colab – Google Colabatory is an executable notebook in which you can program and execute code. It is similar to the Project Jupyter and stores your files in Google Drive. Colab is extremely user friendly. Colab uses a cloud-based runtime.

Development / Architecture

In machine learning you develop and algorithm to solve a problem. However, compared to traditional software development methods you don’t feed in the algorithm and expect the program to find the output. Instead, we feed in the input and output parameters and let the neural networks figure out the relationship between these two. This output will be the algorithm. Neural networks have stacks of layers where each layer has some predefined math and internal variables called weights and biases.



Machine learning models can be very complex or simple. They are created with very different configurations based on the problem you are trying to solve.

Dataset

The dataset was gathered the ISIC archive (The International Skin Imaging Collaboration, 2019). The ISIC archive is a public archive that contains over 200000 images of different types of cancer. Features can be downloaded in large batches via the download tool provided on the website. A search functionality is provided to download images with a specific criterion. Features are downloaded in the JPG format with dimensions of 1022 by 767 pixels. A total of 2000 features are downloaded for each condition. And these are split into two categories. 80% of the features are used for training the data while 20% are used for testing/validation.

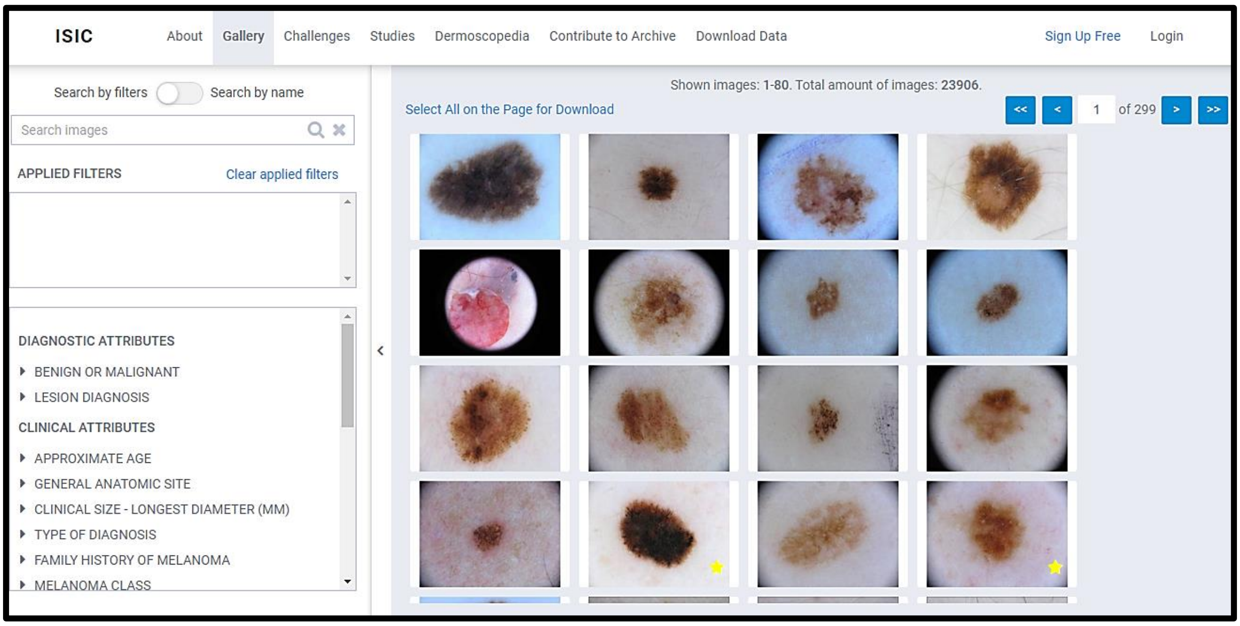


Figure 1 - Dataset Download Tool

Model Structure

I will be using the CNN architecture due its numerous advantages with image detection.

```
def create_model():

model = tf.keras.models.Sequential([

    tf.keras.layers.Conv2D(16, (3,3), activation='relu',
input_shape=(IMG_SHAPE,IMG_SHAPE, 3)),

    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2, 2, stride = 2),

    tf.keras.layers.Dropout(0.2),

    tf.keras.layers.Flatten(),

    tf.keras.layers.Dense(512, activation='relu'),

    tf.keras.layers.Dense(5, activation='softmax')

])
```

For my model I have used 4 different types of layers:

- Dense Layer (Processing) – These are layers that are fully connected to one another. This means the neurons are completely linked to the neurons in the preceding layer which is the input layer. The activation function used in this layer is ReLU.
- Convolutional Layer – The traditional convolutional layers use varying number of filters with a size of 3 by 3.
- Pooling Layer – This layer is needed to perform convolutions on RGB images. It reduces the size of the image – down sampling. A 2 by 2 layer with a stride of 2
- Flatten Layer – A flatten layer is used to convert a 2-Dimensional Image into a vector.
- Dropout Layer – A dropout layer was used as a mean to prevent overfitting. It turns off certain neurons, so others can pick them up.
- Dense Layer (Output) – The output layer uses the SoftMax activation function. This is because we want a probability of the image matching percentage. For example – if the image matches.

Another alternative approach to this is **Transfer Learning**. Instead of creating all the layers from scratch we can import a pretrained model that was built and tuned by experts. This means the model has been trained on a large dataset. Simply the learning of an existing model can be transferred to a new dataset. To perform transfer learning, we must change the last (output) layer of the model to match the output classes of the dataset I am using. In this case the output classes would be five, since my model will predict 5 different medical diagnosis. An AI specifically created to detect cancer at Stanford University could be used to transfer learning .

Activation Functions

As seen on the code above I have used two types of activation functions. The Rectified Linear Unit has been used in the identification of images alongside the convolutional layers. The reason for my choice is that it can be used to avoid the problem on vanishing gradient. And the SoftMax functions is used in the final dense layer to produce an output that will be a probability.

Compilation

When compiling the model two parameters are defined. The Loss and Optimizer functions.

- The Loss function measures the difference between the input and output. There are various types of loss functions. One of the most commonly used classification loss function is the Binary Cross Entropy, which I used. Some of the factors that affect the choice of Loss functions are the presence of outliers in the data, the classification algorithm and the gradient descent. The Binary Cross Entropy is the better choice from the list of classification Loss functions because the function measures the performance of a classification model where the output will be a probability between 1 and 0. In my model the output images will be ranked on a percentage match or likeliness.

$$H(p,q) = -\sum_x p(x)\,\log q(x).$$

Figure 2 - Formula for calculating Cross Entropy Loss

- The Optimizer function adjusts the internal variables as much to reduce the loss and produce faster and better results. There are two types of Optimizer algorithms. Namely the Constant Learning Rate and Adaptive. The **Constant Learning Rate** is the most popular and used among the two and uses the gradient descent.

$$W^{(k+1)} = W^{(k)} - \eta * (\Delta J(W))$$

Figure 3 - Constant Learning Rate Formula

The learning rate is denoted by **η**. This needs tuning to create the perfect algorithm. If the **η** value is too little it can cause the increasing the time consumed to train the model. However, if the value of **η** is too big the loss function may vary. There should be a perfect value between these for the model is perform well.

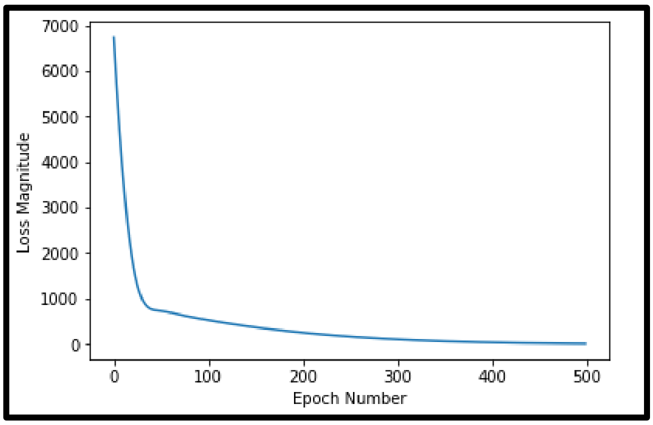
There may be various problems when using the gradient descent such as the **η** value needs to be outlined up front. Another problem that occurs is that the parameters will be applied unified. This is the reason I have chosen to use Adaptive learning, more specifically the Adam

(Adaptive Moment Estimation) optimizer. The Adam optimizer is more practical and outperforms adagrad and gradient descent. To judge how well my model performs I have set the metrics to accuracy rather than loss. This shows the probability of how accurate my model was when it produced a result.

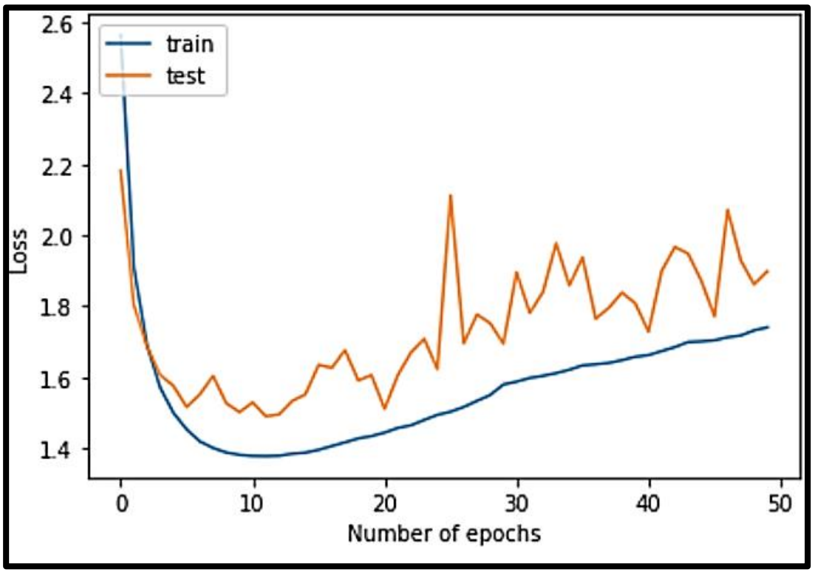
```
# Compile the model
model.compile(optimizer='adam(0.1)',
          loss='sparse_categorical_crossentropy',
          metrics=['accuracy'])
```

Training

An epoch is the number of times an AI trains. To get accurate results an AI will train for thousands or millions of iterations. When epochs increased, the loss is reduced. This is because the AI will improve with every epoch. The weights and biases are tuned until it best matches the output. This can be seen by visualizing the graph.



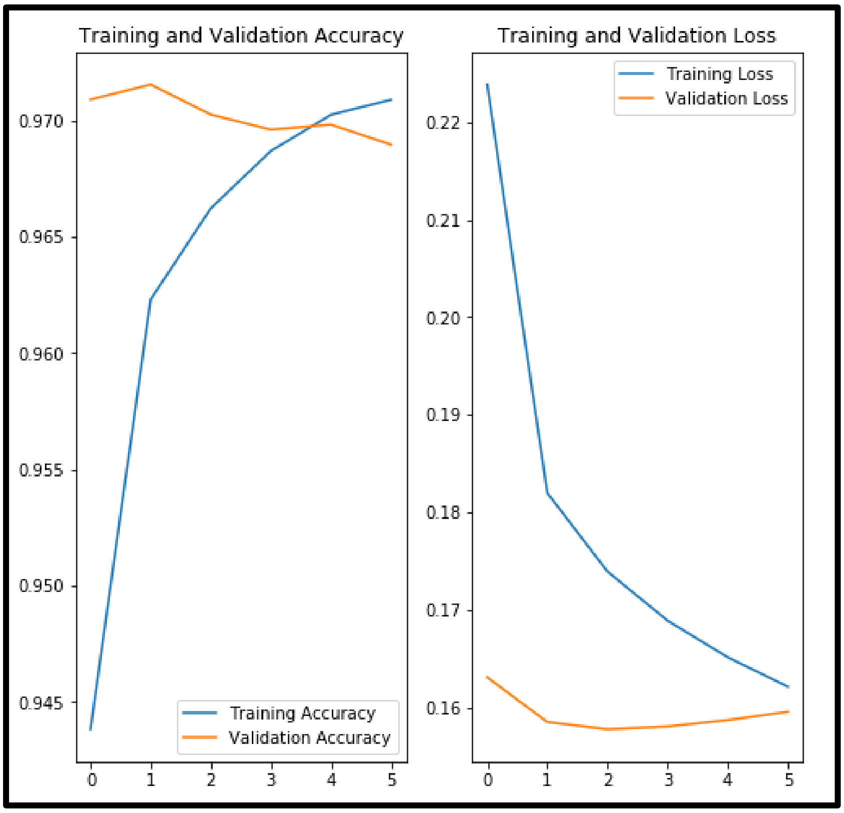
However, this will not always be true. When I trained the model, it started to overfit. **Overfitting** is when the model after a certain number of epochs of training, it starts to memorize the dataset. This means that the model will no longer improve or even sometimes become worse.



As it can be seen here, the model performs very poorly causing the loss rate to increase significantly after about 10 epochs.

There are various methods I used to prevent overfitting. Such as:

- Data Augmentation – Rotation, zooming and resizing of images.
- Large Dataset – A large dataset can prevent overfitting.
- Early Stopping – In early stopping we introduce a new dataset called the validation set. This is used to compare the training loss whole training. We can also put a constraint on the weights and biases or checkup on the dataset error occasionally during training. If it goes up it's time to stop training.
- Dropout Layer – A dropout layer turns off certain neurons randomly throughout the training process. This in turn increases accuracy and prevents the model from memorizing the data as other neurons will have to pick up the work. As seen on the code above I set a dropout of 20%.



The graph demonstrates the validation loss and accuracy after applying the methods to prevent overfitting. As seen there is a drastic improvement on the performance of the model. The training loss falls continually, and the validation loss only increases by a very small margin.

Similarly, the training accuracy increases continually while the validation accuracy falls by a tiny margin when compared to the previous graphs.

Below I have trained my model for 50 epochs.

```
epochs = 50
history = model.fit_generator(
    train_data_gen,
    steps_per_epoch=int(np.ceil(train_data_gen.n / float(batch_size))),
    epochs=epochs,
    validation_data=val_data_gen,
    validation_steps=int(np.ceil(val_data_gen.n / float(batch_size)))
)
```

Testing

An image that represents a disease is captured using the devices camera. This is pulled into the program and the predict model command runs the image through the model to check if it compares as a disease.

```
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():

  print('The file "{name}" has been uploaded with length {length} bytes'.format(

      name=fn, length=len(uploaded[fn])))

print(model.predict(ISIC_0000000.jpg)

print("Labels: ", label_batch)

print("Predicted labels: ", predicted_ids)
```

AI TECHNIQUES IN MEDICAL DIAGNOSIS

There are three main machine learning paradigms: supervised, unsupervised and reinforcement learning. For my model I have chosen **supervised learning**. The model will create certain parameters while training which will be used to map the inputs to outputs. Then the model is exported into an application and run on a new environment.

There are various AI techniques such as: Natural Language Processing (NLP), Decision Tree (DT), Support Vector Machines (SVM), etc.

The task at hand is image classification. There are two AI techniques that are well suited for this task. Artificial Neural Networks (ANN) and Support Vector Machines (SVM).

Neural networks are the best methods when patterns start to get complex. For a simple pattern SVM (Support Vector Machines) are just about enough. But in complex situations neural networks start to outperform older methods. SVM's perform poorly when the number of features is greater than the number of samples. SVM's do not directly provide probability estimates. These should be performed separately with Cross-Validation.

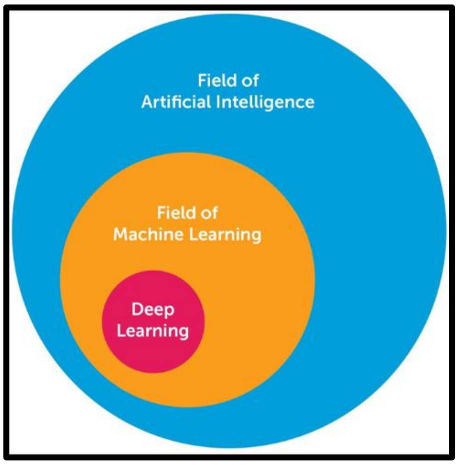
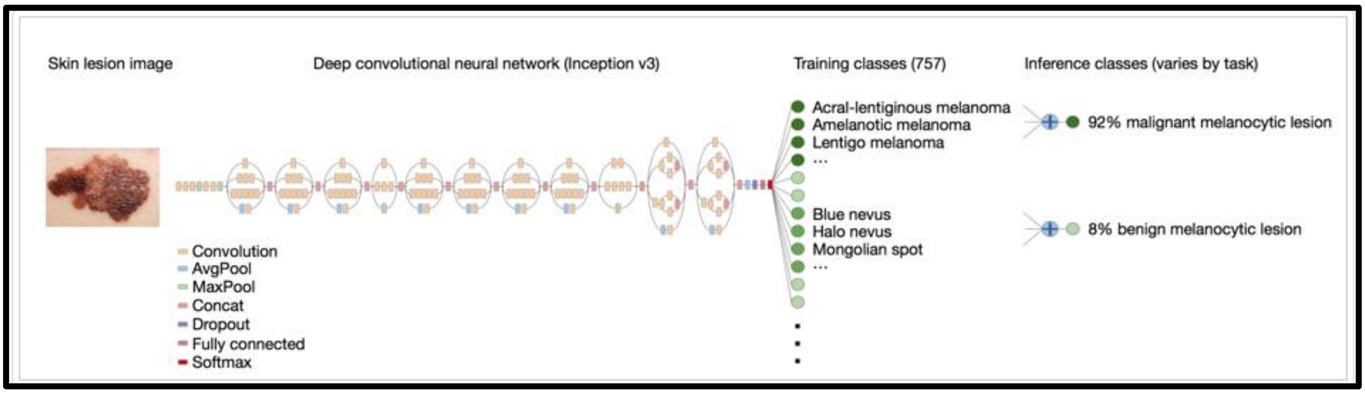


Figure 4 - Deep Learning and Machine Learning are a subset of AI

Artificial Neural Networks look like our brain. Nodes that resemble neurons in our brain are interconnected to one another.

Among Artificial Neural Networks, Convolutional Neural Networks perform the best image classification being able to reach human level accuracy. They are great when dealing with invariances such as lightings and angles.

Sebastian Thrun who is a professor at the Stanford University is also the lead of an AI that was developed to detect skin cancer. The model beat 21 certified dermatologists with a staggering accuracy of 91%. This was made possible by using the Inception v3 model. Below is the structure of that model.



The flow of the data is from the left to right. With a deep Convolutional Neural Network architecture.

My model is similar, but it will be trained to even more types of cancer such as breast cancer and other prevailing medical conditions as well.

Advantages of Artificial Neural Networks

- It's a model that is nonlinear and which makes it easier to comprehend and use when compared to traditional statistical methods.
- Restrictions on input variables are lifted.
- Higher prediction accuracy.

Disadvantages of Artificial Neural Networks

- High powered hardware is needed to train the model
- Unexplainable behaviour – certain problems will be created. Certain solutions will be created without any explanations.

THEORY BEHIND THE AI TECHNIQUE USED

Convolutional Neural Networks (CNN)

Convolutional Neural Networks are the most popular type of deep learning architecture. CNN are now able to detect images better than humans. As medical diagnosis is based on the capture and analysis of images, CNN would be the most appropriate model. CNNs have multiple types of layers and the first is called a convolutional layer. In this layer multiple filters are used simultaneously. Each filter is scanning a different part of the image. The entire convolutional layer is a three-dimensional grid. The convoluted input is then sent to the next layer for activation. The activation function most commonly used in CNN's are **Rectified Linear Unit (ReLU).** CNN's use backdrop for training.

Advantages of CNN's

- The error rate of image recognition has dropped significantly.
- Mechanically senses significant features without human administration.
- Can run on any device – CNN uses a distinct convolution and pooling processes and completes parameter sharing.

Disadvantages of CNN's

- By using CNN, it could take a long time to train the neural net because CNN is highly dependent on the amount and quality of data it is fed. This could also increase cost.

OTHER METHODS

Generative Adversarial Networks (GAN)

Generative Adversarial Network was found by Ian Goodfellow in 2014. Two deep learning models are trained simultaneously. The **generator** tries to create new instances whereas the **discriminator** model tries to identify if the instance is from the training data or the generator. The discriminator in GAN offers direction to the generator on generating images from scratch. The discriminator looks at real images from the dataset and compares the to the images generated. Then a feedback is sent, and the process continues until the generator closely resembles or matches the images from the dataset. Below is an example of how the architecture works.

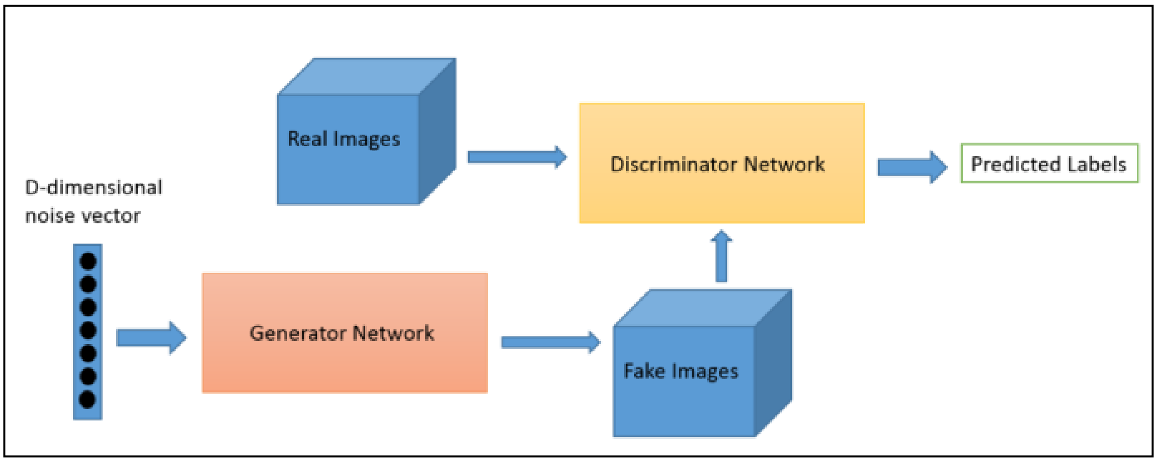


Figure 5 - GAN Face Recognition

Advantages of GAN's

- Since the model has improved over time it is hard to find a difference between generator images or the originals.
- Efficient training.

Disadvantages of GAN's

- Time extensive – Since there are two separate models to train (Discriminator and Generator) training the system would take time.
- Both models need to be working efficiently for the model to succeed.

Residual Networks (ResNets)

CNN are the best choice when it comes to image classification. However, when the problem becomes complex and it becomes hard to train the neural network. There needs to be extra deep layers added to maintain the accuracy of the model. This is where ResNets come into place. Residual Networks have residual modules where each module is a layer. The layers have functions which should be operated on the input.

Advantages of ResNets

- Accuracy. Requires less weights compared to LSTMs and RNNs
- The structure is modular. This is easier to maintain and adapt. It also allows addition of new layers easily.

Disadvantages of ResNets

- When the layers get too deep, detecting error can be extremely difficult

REFERENCES

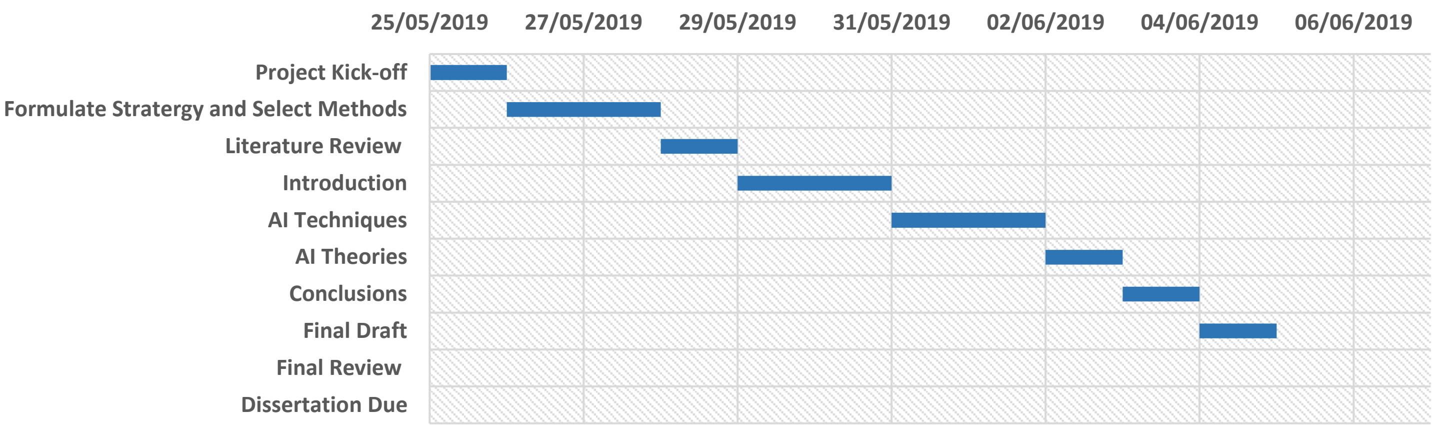
BIBLIOGRAPHY

The International Skin Imaging Collaboration. (2019, May). Retrieved from ISIC Archive: https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery

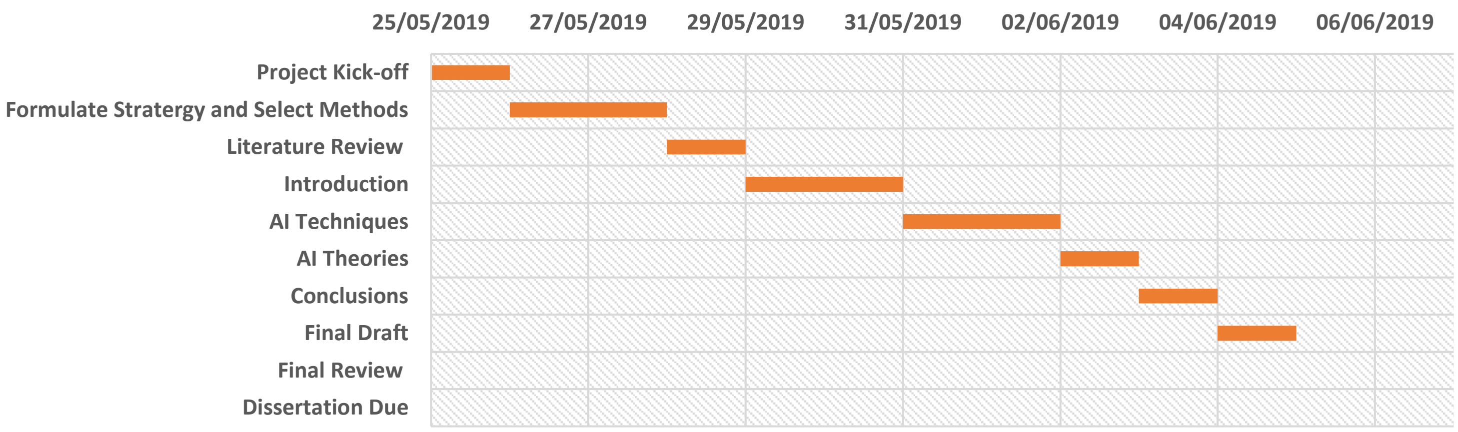
APPENDIX

Gantt Chart

1. Planned Gantt Chart



2. Actual Gantt Chart



PLAGARISM REPORT

Cancer Detection - Mushad Aadil.pdf

ORIGINALITY REPORT

29%	8%	3%	29%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Wales Institute, Cardiff Student Paper	22%
2	Submitted to Columbia University Student Paper	1%
3	Dattaraj Jagdish Rao. "Keras to Kubernetes®", Wiley, 2019 Publication	1%
4	Submitted to University of Portsmouth Student Paper	1%
5	intelligence-artificielle.agency Internet Source	1%
6	Submitted to University of Liverpool Student Paper	1%
7	Lina Marlina, Cipto Wardoyo, W.S. Mada Sanjaya, Dyah Anggraeni, Sinta Fatmala Dewi, Akhmad Roziqin, Sri Maryanti. "Makhraj recognition of Hijaiyah letter for children based on Mel-Frequency Cepstrum Coefficients	<1%

This higher percentage is due to the repetitive format used, such as the cover page, contents page, headers and footers. CMU was previously named as the university above, hence detected from my previous papers. For more information refer to the Similarity Report on Turnitin.

(MFCC) and Support Vector Machines (SVM) method", 2018 International Conference on Information and Communications Technology (ICOIACT), 2018
Publication

8	Submitted to University of Macau Student Paper	<1%
9	www.mtc.edu.eg Internet Source	<1%
10	Submitted to University of Hertfordshire Student Paper	<1%
11	Submitted to Chandigarh University Student Paper	<1%
12	"Machine Learning in Medical Imaging", Springer Science and Business Media LLC, 2017 Publication	<1%
13	Submitted to Asian Institute of Technology Student Paper	<1%
14	Marwan Ali Albahar. "Skin Lesion Classification Using Convolutional Neural Network With Novel Regularizer", IEEE Access, 2019 Publication	<1%
15	Submitted to University of Lincoln Student Paper	<1%

16	Submitted to Sabanci Universitesi Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		