**Machine Learning for Medical Image Analysis and Classification.**

**Kingston University London| MSc Data Science| Course Work2 | Group Number:24**

**Team Members: Bhavana Koppula - K2278142| Jickson Dijo Mathew - K2258245| Bhavana Raj Nagaraj Srinivasappa - K2267997| Darpan Sanjay Agarwal - K2279745**

1. **Aims and Objectives of your theme problem:**

MEDMNIST V2 is a standardised biomedical images given with group of 2D and 3D datasets. Primary aim of Machine Learning for Medical Image Analysis and Classification project is to develop machine learning algorithms with deep neural networks using Python and Keras for BloodMNIST, BreastMNIST 2D datasets which are subsets of MEDMNIST(details are available in <https://medmnist.com/>). Datasets in the format .npz is downloaded from Zenoda site[Files: [bloodmnist.npz](https://zenodo.org/record/6496656/files/bloodmnist.npz?download=1), [breastmnist.npz](https://zenodo.org/record/6496656/files/breastmnist.npz?download=1)].Images are given with the size of 28 × 28 (2D) and class labels for each dataset is taken from the Git repo dataset.py .

The process of designing and deploying deep neural networks are followed by loading the BloodMNIST and BreastMNIST datasets from the MedMNIST2D and verifying the number of records from training and test samples. Normalize and format the data for multidimensional/binary classification and visualize a subset of the data and ground truth. Data pre-processing, normalization, augmentation ,reshape and resize are performed on the datasets as per the model requirements. For imbalanced data necessary data augmentation and data generation carried out on the training set. Designed and trained two deep neural network models for BloodMNIST and BreastMNIST datasets respectively, resulting in four separate models. Model evaluation is performed on each datasets to calculate the F1-Score,Recall,Precision,Accuracy and Confusion metrics.

The deep learning models will be evaluated using various metrics and a comparative study will be conducted on all models by presenting tabular comparative report. Finally website is developed using streamlit and python to present the user interface to load the .npz files. Model selection for individual dataset can be performed by the user using dropdown there by deploying the model and training the datasets. User interface will display the details related to data loading, visualization, model summary, confusion matrix, Evaluation Scores and Image predictions of the built model.

More details about Dataset shown below shown from scientific data paper given in MEDNMIST site

* ***BreastMNIST****.* This data set is on breast ultrasound images. Class labels provided in git as follows "malignant","normal, benign”. Source dataset is split into 7:1:2 into training, validation and test set. The source images resized into 1×28×28.
* ***BloodMNIST****.* This dataset contains 17,092 images on individual normal cells captures on various conditions. Eight Class labels provided in git as follows basophil,eosinophil,erythroblast,immature granulocytes(myelocytes, metamyelocytes and promyelocytes),lymphocyte,monocyte,neutrophil,platelet.Source dataset is split into 7:1:2 into training, validation and test set. The source images resized into 3×28×28.

1. **Analysis of the existing DNNs:**

Several bench mark deep neural networks built for MEDMNIST 2D datasets. Below are the following types of models built.

* ResNets : ResNets with early stopping strategy are built with following specifications. Four Residual networks, few layers of convolutional layers, batch normalization, relu activation, input channel as 3,model training as cross entropy-loss, batch size as 128,100 epochs, Adam optimiser with initial learning rate as 0.001
* Auto-Sklearn: It is an open source AutoML method for statistical learning. This method automatically searches the algorithms and hyper parameters in scikit-learn. Images are flattened into one dimensional data and corresponding labels used for fitting the model.
* Auto Keras: It is an open source AutoML method which searches deep neural networks and hyper-parameters.For each dataset, model tries 20 Keras models and trains each model for 20 epochs. Highest AUC score is considered for evaluation.
* Google AutoML Vision: It is an tool offered by Google. Edge exportable models are trained for 2D datasets on google ML platform and export into tensor lite for offline interference.

**Table

Description automatically generated**

Fig: Evaluation Metrics for 2D BLOODMNIST and BREASTMNIST data set.

**Evaluation:**

Google AutoML Vision outperforms all the other methods in average AUC, however, it is very close to the performance of baseline ResNets. The ResNets surpass auto-sklearn and AutoKeras, and outperform Google AutoML Vision in average ACC. Under the same backbone, the datasets with resolution of 224 win higher AUC and ACC score than resolution of 28. While under the same resolution, ResNet-18 is superior to ResNet-50.

1. **Analysis of the designed DNNs:**

In the project four deep neural networks have been designed and deployed for medical image processing of 2D datasets. CNN and Resnet50 are designed for the bloodmnist which is multiclass image classification problem. Resnet101 and InceptionV3 are designed for the breastmnist dataset which is binary image classification problem. As per the necessity of each model data pre-processing, data normalisation, data augmentation, data reshaping, data resizing, data generation has been implemented.

**Blood Dataset DNN Design:**

* **CNN (Convolutional Neural Network)**

Convolutional Neural Networks are made up of neurons that have learnable weights and biases Each neuron receives some inputs, performs a dot product, and optionally follows it with a non-linearity.

Before training the model below operations are performed on data,

* Training, validation ,Test samples and respective class labels are segregated as per the keys.
* Data visualization plotted for few sample images with class.
* Transforming training data into float
* Converting our values from range[0,255] to range[0,1], to normalizes the image
* Transforming y label data to categorical
* Input size is 28\*28 and channel is 3

The network used in the CNN model building is sequential, since the BloodMNIST belong to medmnist2D conv2D is the layer used to 2D image processing to produce a set of output features. This code adds a Conv2D layer to the model with 32 filters of 3x3 size, using the RELU activation function. The shape of input parameter specifies the dimensions of the input image i.e,28x28x3. By default, the stride is set to (1, 1) and the padding is set to 'same'. To reduce the risk of overfitting maxpooling2D layer is used. Input shape of the pooling layer is automatically inferred from the output shape of the previous layer. The default stride for pooling is the same as the pool size i.e,2x2. During training, 25% of the inputs to the dropout layer will be randomly set to 0, helping to prevent overfitting. The code adds a Flatten layer to the model after the convolutional layer. The Flatten layer used to flatten the output of the convolutional layer into a 1D vector, which is then passed to a fully connected layer with 512 units and RELU activation. the Dense layer with SoftMax activation is added as the output layer of the neural network, with units=8, indicating that there are 8 output classes which can be used to make predictions on new data.

Diagram

Description automatically generated

Fig: Convolutional Neural Network Architecture

* **RESNET50:**

ResNet50 is a pre-defined convolutional neural network architecture in Keras that was introduced in the research paper "Deep Residual Learning for Image Recognition" by He et al. (2015).This architecture has residual nets with a depth of up to 152 layers.

Before training the model below operations are performed on data,

* Training, validation ,Test samples and respective class labels are segregated as per the keys.
* Data visualization plotted for few sample images with class.
* Transforming training data into float
* Change image shape to 32,32
* Converting our values from range[0,255] to range[0,1], to normalizes the image
* Transforming y label data to categorical
* Pre-processing training data
* Input size is 32\*32 and channel is 3

The parameters used in RESNET50 model for include\_top as false i.e., only the convolutional layers are included, and the model can be used as a base for building a new model. The weights are set to ImageNet. Intermediate Dense layer with 256 layers and activation is considered as Relu after performing flattening. The output layer has an activation function of SoftMax and 8 units. Now the base layers have been frozen and then training can be started later on. Freezing the layers means that their weights will not be updated during training, which is useful when you want to use a pre-trained model as a feature extractor.

Chart

Description automatically generated

Fig: RESNET50 architecture

**Breast Dataset DNN Design:**

As the breast dataset is small data set , Data augmentation and data generation has been performed on the training data set using the function ImageDataGenerator().This function automatically turns image files into pre-processed tensors to feed directly into models. Image transformation operations are applied on the training dataset such as rotation, translation and zooming to produce new versions of existing images. We input these new images into our model to get the better accuracy.

* **INCEPTION V3:**

ResNet50 is a pre-defined convolutional neural network architecture in Keras that was introduced Rethinking the Inception Architecture for Computer Vision.

Before training the model below operations are performed on data,

* Training, validation ,Test samples and respective class labels are segregated as per the keys.
* Data visualization plotted for few sample images with class.
* Transforming training data into float
* Converting the grayscale image to RGB
* Change image shape to 75,75
* Converting our values from range[0,255] to range[0,1], to normalizes the image
* Transforming y label data to categorical
* Pre-processing training data
* Input size is 75\*75 and channel converted is 3

Chart, box and whisker chart

Description automatically generated

Fig: INCEPTIONV3 architecture

The parameters used in INCEPTIONV3 model for include\_top as false i.e., only the convolutional layers are included, and the model can be used as a base for building a new model. The weights are set to ImageNet. Now freeze weights of all layers till except the last 10. The "mixed7" layer is the seventh mixed convolutional layer in the Inception architecture. Batch normalisation is performed adding to the previous layer. Followed by Dense 1024 layers with Relu activation and drop out added as 0.2. The output layer has an activation function of sigmoid and 2 units.

* **RESNET101:**

ResNet50 is a pre-defined convolutional neural network architecture in Keras that was introduced Rethinking the Inception Architecture for Computer Vision

Before training the model below operations are performed on data,

* Training, Validation ,Test samples and respective class labels are segregated as per the keys.
* Data visualization plotted for few sample images with class.
* Transforming training data into float
* Zoom the image by 0.1by using the parameter zoom\_range.
* Rotate the image as 50degree or 25 degree in different augmentation sections using the parameter rotation\_range
* Change the horizontal and vertical images by 0.2 and 0.1 factors as per different sections by using width\_shift\_range and the height\_shift\_range parameters.
* Apply shear-based transformations using the shear\_range parameter
* Set Vertical and Horizontal flip parameters to True
* Increasing the training and train label dataset based on data augmentation.
* Converting the grayscale image to RGB
* Change image shape to 32,32
* Converting our values from range[0,255] to range[0,1], to normalizes the image
* Transforming y label data to categorical
* Pre-processing training data
* Input size is 32\*32 and channel converted is 3

Diagram

Description automatically generated

Fig: RESNET101 architecture.

The parameters used in RESNET101 model for include\_top as false i.e., only the convolutional layers are included, and the model can be used as a base for building a new model. The weights are set to ImageNet. Now freeze the base layer models with trainable parameter as False. Followed by Dense 256 layers with Relu activation. The output layer has an activation function of sigmoid and 2 units.

1. **Analysis of the training process**

For Blood and Breast datasets has been trained on epochs and batch sizes as per below table. For multi class classification problem, Loss is considered based on categorical cross entropy to calculate the difference between predicted probability and true probability of the classes. For Binary class classification problem, Loss is considered based on Binary cross entropy. Optimizer used for all the models is ‘Adam’ as it updates the weights of neural networks during training which results in increased accuracy.

**Table

Description automatically generated**

1. **Analysis of the designed interactive application and the related GUI**

From streamlit the graphical user interface has been built for this project to display the complete applications.

**Requirements:**

* Google Colab
* Streamlit
* Google Drive
* Files uploaded on Box
* GPU

**GUI overview:**

* Install the required packages
* Connect file to google drive
* Upload datasets, images and neural network models from box folder to google drive
* Check for the correct path from the drive to upload the models
* GUI structure:
  + User will first upload the image they want to make a classification of.
  + The user will be able to see the image they have uploaded.
  + The image will be captured by Py file and it will ask user for an option between blood and breast.
  + Once user selects the category two models will run automatically
  + The model summary and individual result will be displayed
  + The Py file will automatically return the best model amongst the two to the user with potential class label and likelihood of it.
  + The likelihood range is 0 to 1 ,where 1 is the most accurate prediction.
  + Deleting the image will reset the page and user can try another image.

1. **Comparative analysis and performance evaluation of all the DNNs used**

The DNN models used evaluation metrices like F1 score, accuracy , precision, recall and confusion matrix. precision and recall matrices are used for predicting performance. Precision is measure of similarity of outcome in information retrieval, recall is how often the genuinely valid items are retrieved. Since the dataset is not balanced f1 score is as good as accuracy scoref1 score is same as harmonic mean of precision and recall.

* **Confusion Metrics of DNN models:**

Below figure shows the confusion metrics representation of DNN models designed for Blood and Breast datasets.

**BloodMnist:**

Calendar

Description automatically generated A picture containing text, keyboard

Description automatically generated

Fig:CNN[1] Fig:RESNET50[2]

**BreastMnist:**

A black background with white text

Description automatically generated with low confidence Graphical user interface, text, application

Description automatically generated

Fig:INCEPTIONV3[1]. Fig:RESTNET101[2]

* + **Evaluation metrics of DNN models comparison analysis**

For BlOODMNIST dataset, CNN model accuracy of 87.35 percentage is performing better than compared to RESNET50.Rest of the evaluations scores of CNN is better than compared to RESNET50.

For BREASTMNIST dataset, both transfer learning models are designed. INCEPTIONV3 with accuracy 77.31 percent is performing better than compared to RESNET101. Rest of the model parameters are performing well for INCEPTION V3.

A screenshot of a computer

Description automatically generated with low confidence

1. **References:**

* *The structure of the ResNet-101-based deep feature extractor.* (no date). Available at: https://www.researchgate.net/figure/The-structure-of-the-ResNet-101-based-deep-feature-extractor\_fig4\_348078198 (Accessed: March 13, 2023).
* *Performance evaluation of deep CNN-based crack detection ... - researchgate* (no date). Available at: https://www.researchgate.net/publication/349717475\_Performance\_Evaluation\_of\_Deep\_CNN-Based\_Crack\_Detection\_and\_Localization\_Techniques\_for\_Concrete\_Structures (Accessed: March 13, 2023).
* *Human gender classification using transfer learning via pareto frontier ...* (no date). Available at: https://www.researchgate.net/publication/340614544\_Human\_Gender\_Classification\_Using\_Transfer\_Learning\_Via\_Pareto\_Frontier\_CNN\_Networks (Accessed: March 13, 2023).
* *MedMNIST*. Available at: https://medmnist.com/ (Accessed: March 13, 2023).
* Yang, J. *et al.* (2023) *MedMNIST v2 - a large-scale lightweight benchmark for 2D and 3D Biomedical Image Classification*, *Nature News*. Nature Publishing Group. Available at: https://www.nature.com/articles/s41597-022-01721-8 (Accessed: March 13, 2023).
* MedMNIST (no date) *Experiments/medmnist2d at main · MedMNIST/Experiments*, *GitHub*. Available at: https://github.com/MedMNIST/experiments/tree/main/MedMNIST2D (Accessed: March 13, 2023).
* Team, K. (no date) *Keras Documentation: Keras applications*, *Keras*. Available at: https://keras.io/api/applications/ (Accessed: March 13, 2023).
* He, K. *et al.* (2015) *Deep residual learning for image recognition*, *arXiv.org*. Available at: https://arxiv.org/abs/1512.03385 (Accessed: March 13, 2023).
* Gautam, T. (2021) *Image classification in python with keras: Image classification*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2020/10/create-image-classification-model-python-keras/ (Accessed: March 13, 2023).
* Hershdhillon (no date) *CNN-and-VGG-16-model-implementation-on-fashion-mnist/fashion\_mnist\_vgg16\_implementation.ipynb at master · Hershdhillon/CNN-and-VGG-16-model-implementation-on-fashion-MNIST*, *GitHub*. Available at: https://github.com/hershdhillon/CNN-and-VGG-16-Model-Implementation-on-Fashion-MNIST/blob/master/Fashion\_MNIST\_VGG16\_Implementation.ipynb (Accessed: March 13, 2023).
* Thakur, R. (2020) *Step by step VGG16 implementation in Keras for beginners*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c (Accessed: March 13, 2023).
* 262588213843476 (no date) *Deeplearning-ai/Mnist-vgg.ipynb*, *Gist*. Available at: https://gist.github.com/nikogamulin/c0892e33ff823e1cb45caaaf5a949fd1 (Accessed: March 13, 2023).
* Donatastamosauskas (2018) *Using ResNet for mnist*, *Kaggle*. Kaggle. Available at: https://www.kaggle.com/code/donatastamosauskas/using-resnet-for-mnist (Accessed: March 13, 2023).
* Lizhecheng (2023) *RESNET\_BASED\_CNN intel\_image\_classification*, *Kaggle*. Kaggle. Available at: https://www.kaggle.com/code/lizhecheng/resnet-based-cnn-intel-image-classification (Accessed: March 13, 2023).
* Iafoss (2018) *Pretrained Resnet34 with RGBY (0.460 public LB)*, *Kaggle*. Kaggle. Available at: https://www.kaggle.com/code/iafoss/pretrained-resnet34-with-rgby-0-460-public-lb (Accessed: March 13, 2023).
* Akshatapatel (2019) *Breast-cancer-image-classification/resnet.ipynb at master · akshatapatel/breast-cancer-image-classification*, *GitHub*. Available at: https://github.com/akshatapatel/Breast-Cancer-Image-Classification/blob/master/ResNet.ipynb (Accessed: March 13, 2023).
* Victor Zhou (no date) *Keras for beginners: Implementing a Convolutional Neural Network*, *Victor Zhou*. Victor Zhou. Available at: https://victorzhou.com/blog/keras-cnn-tutorial/ (Accessed: March 13, 2023).
* Sachin (2020) *Cats-vs-dogs*, *Kaggle*. Available at: https://www.kaggle.com/datasets/shaunthesheep/microsoft-catsvsdogs-dataset (Accessed: March 13, 2023).
* *How to build a Streamlit app (beginner level Streamlit tutorial) - part 1* (2020) *YouTube*. YouTube. Available at: https://www.youtube.com/watch?v=-IM3531b1XU (Accessed: March 13, 2023).
* DiRenzo, N. (2022) *Deploying ML models using Streamlit*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/deploying-ml-models-using-streamlit-5d6212453bdd (Accessed: March 13, 2023).
* Early detection of skin cancer using Deep Learning ... - IEEE xplore (no date). Available at: https://ieeexplore.ieee.org/document/8972045/ (Accessed: March 13, 2023).
* Vectorig (no date) Earth abstract connection., iStock. Available at: https://www.istockphoto.com/vector/graphic-of-dots-connected-by-lines-in-blue-and-white-tones-gm455445559-23246232 (Accessed: March 15, 2023).
* Source code from lecturing session conducted
* Imshow (no date) Imshow in Python. Available at: https://plotly.com/python/imshow/ (Accessed: March 15, 2023).
* Paper, D. (no date) State-of-the-art deep learning models in tensorflow, SpringerLink. Apress. Available at: https://link.springer.com/book/10.1007/978-1-4842-7341-8 (Accessed: March 16, 2023).
* Inception\_cats\_and\_dogs. Available at: http://www.mommy-scientist.ru/inception\_cats\_and\_dogs.html (Accessed: March 16, 2023).
* Shah, D. (2020) Early fire detection system using deep learning and opencv, Medium. Towards Data Science. Available at: https://towardsdatascience.com/early-fire-detection-system-using-deep-learning-and-opencv-6cb60260d54a (Accessed: March 16, 2023).
* TensorFlow 2.0 computer vision cookbook: Implement machine learning solutions to overcome various computer vision challenges 183882913X, 9781838829131 (no date) ebin.pub. Available at: https://ebin.pub/tensorflow-20-computer-vision-cookbook-implement-machine-learning-solutions-to-overcome-various-computer-vision-challenges-183882913x-9781838829131.html (Accessed: March 16, 2023).
* Simple audio recognition: Recognizing keywords &nbsp;: &nbsp; Tensorflow Core (no date) TensorFlow. Available at: https://www.tensorflow.org/tutorials/audio/simple\_audio (Accessed: March 16, 2023).
* TSCCNN slides - github pages (no date). Available at: https://ivansanchezfernandez.github.io/TSC\_training\_validation/ (Accessed: March 16, 2023).