**REPORT**

**Impact of Image Processing on Deep Learning Models.**

**1.INTRODUCTION**

* This experiment was conducted to compare how deep learning models (specifically those working on image data) perform when input images are not processed by image processing techniques vs how they perform when input images are process by image processing techniques such as contrast stretching, image augmentation, image enhancement, etc.
* The dataset chosen for the experiment has the task of classifying bone marrow cells into separate categories.
* A deep learning architecture was developed and both the scenarios of input images were passed to the same architecture for accurate comparisons.

**2.PROBLEM STATEMENT AND MOTIVATION**

* **Problem Statement**: Create a multi-class classification model to predict cell abnormalities in bone marrow cell images
* Bone marrow cells show whether healthy blood cells are being produced or not. These observations are used by doctors to diagnose and monitor blood diseases, cancers and fevers of unknown origin[1]
* Classification of such cells requires a lot of medical expertise and can be done only by experts in the field. This process can be assisted greatly by an automated system that can classify cells into various categories. The system itself can be amplified in performance by improving the quality of the input images. This was the motivation behind carrying out this experiment

**3.DATASET**

* The dataset contains a collection of over 170,000 de-identified, expert-annotated cells stained using the May-Grunwald-Giemsa/Pappenheim stain[2]
* All samples were scanned and post-processed in the Munich Leukemia Laboratory (MLL).
* For this experiment 6000 images were chosen belonging to 3 classes.
* 4200 images were chosen for the training set, 1800 images for the testing set.
* The 3 classes are as follows:

1.Artefact(ART),

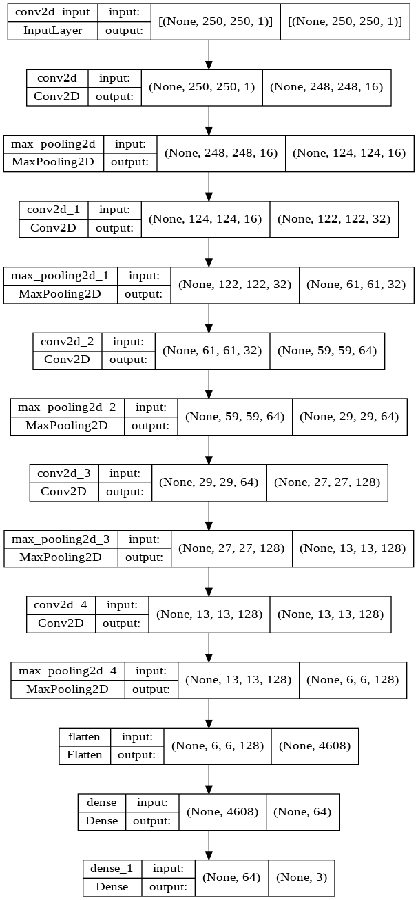
2.Blast(BLA),

3.Lymphocyte(LYT),

* These classes were chosen out of the overall pool as there were substantial images in each category to train the model without developing bias for a particular class. Also, these are the more common types of cell categories or deformities observed.

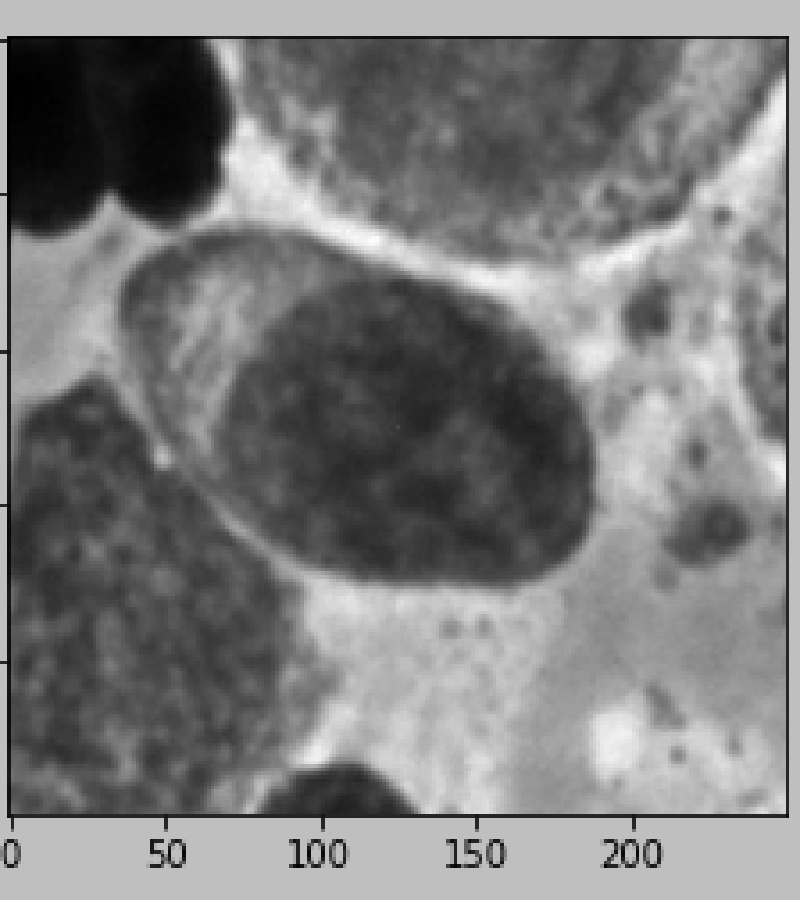
**4.MODEL(ARCHITECTURE OF THE DEEP CONVOLUTIONAL NEURAL NETWORK)**

* A CNN architecture was developed for the classification task.
* The architecture was kept the same for both the cases – training and testing without image processing and training and testing with image processing.
* Two different models were generated as stated above
* The same dataset was input to both the models to assess the results fairly
* The architecture is detailed in the next slide



**5.IMPLEMENTATION WITHOUT IMAGE PROCESSING**

* The training data was loaded into Dataset objects in ‘grayscale’ format by using the Keras function – image\_dataset\_from\_directory.
* The Dataset objects were then converted into NumPy arrays to input to the model
* The training labels were one-hot encoded
* A model was built with loss function as ‘categorical crossentropy’, optimizer as ‘ADAM’
* The training inputs and labels were fed to the model in batch sizes of 100 and a validation split of 0.2. The model was trained for 15 epochs.
* The testing data was loaded using the same function mentioned above
* The testing labels were one-hot encoded
* model.evaluate was called on the testing inputs and testing labels
* *Below is the attached unprocessed image of one of the cells of the bone marrow.*

**

**6.IMPLEMENTATION WITH IMAGE PROCESSING.**

* Image Processing techniques were applied on the original NumPy arrays of training data
* The processed images were stored as NumPy arrays
* The NumPy arrays were fed to a new model for training
* The new model was trained on the processed images
* The exact same processing techniques were applied on the original NumPy arrays of testing data
* The processed testing images were stored as NumPy arrays
* model.evaluate was called on the testing images and testing labels
* The image processing techniques used are explained in the following slides

***Following methdology was followed to preprocess the image using image processing and the pre-processed image was given as input to the deep CNN .***

1.SHARPENING.

2.POWER LAW TRANSFORMATION.

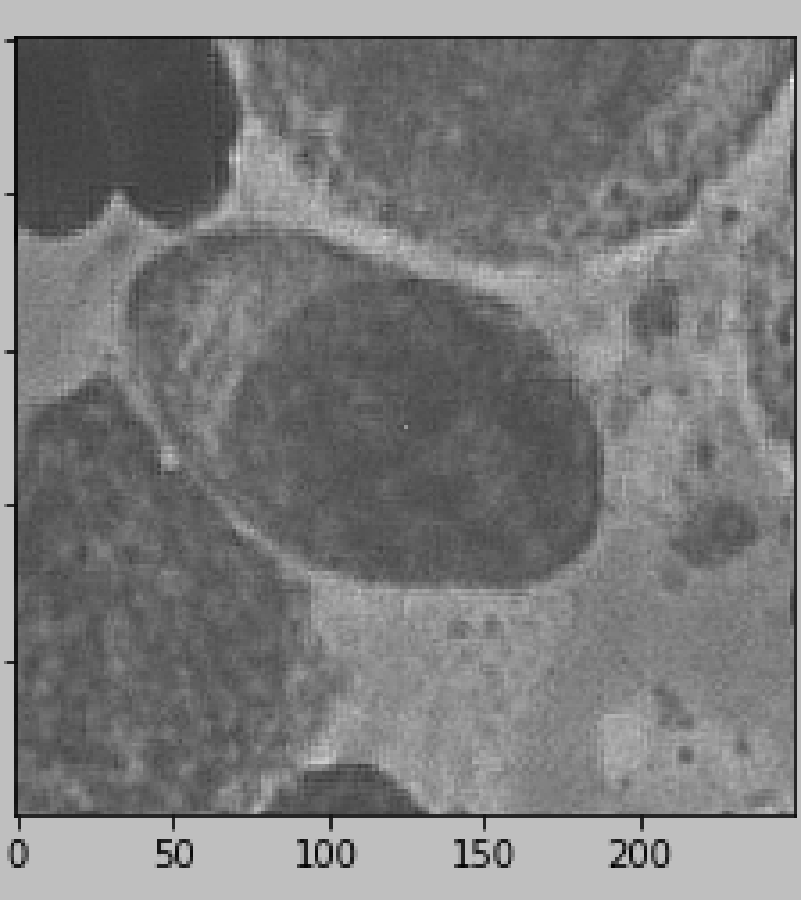
3.IMAGE NORMALIZATION/ RESCALING

1. **IMAGE SHARPENING.**

* Sharpening is an Image Processing technique used to improve the detailing of the image by adding sharpness to it
* A sharpening filter of size sXs is applied on the image which then changes the pixel values in the receptive field respectively
* Sharpening has been implemented in this experiment by using the functionalities of NumPy and OpenCV
* The sharpening filter used in this experiment is shown below:

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 9 | -1 |
| -1 | -1 | -1 |

* *Below shown is one of the sharpened images.*

**

**B.POWER LAW TRANSFORMATION**

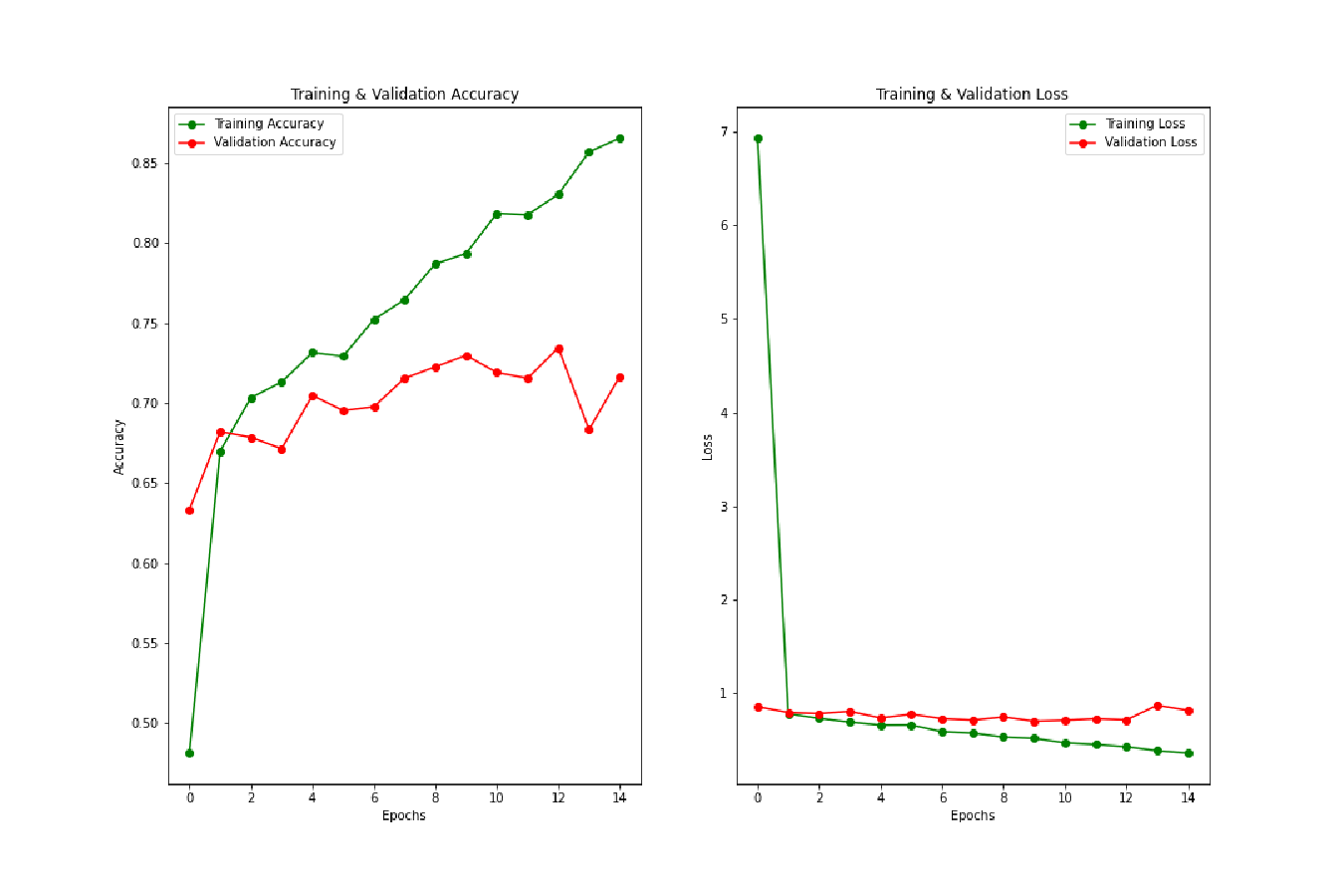
* Power Law Transformation is used for enhancing the display quality of an image by changing its contrast details.
* It is applied on an image ‘s’ by the following equation:
* s = cr^g
* Here c is constant and g is known as gamma
* The pixels in s change their values as per the equation
* Increasing the gamma value introduces more contrast in the image by darkening it
* A lower gamma value will give a less contrasting and more well-lit image
* For this experiment the gamma value has been chosen as 1.6
* *Below shown is the image of cell after*

*Applying power law transformation.*

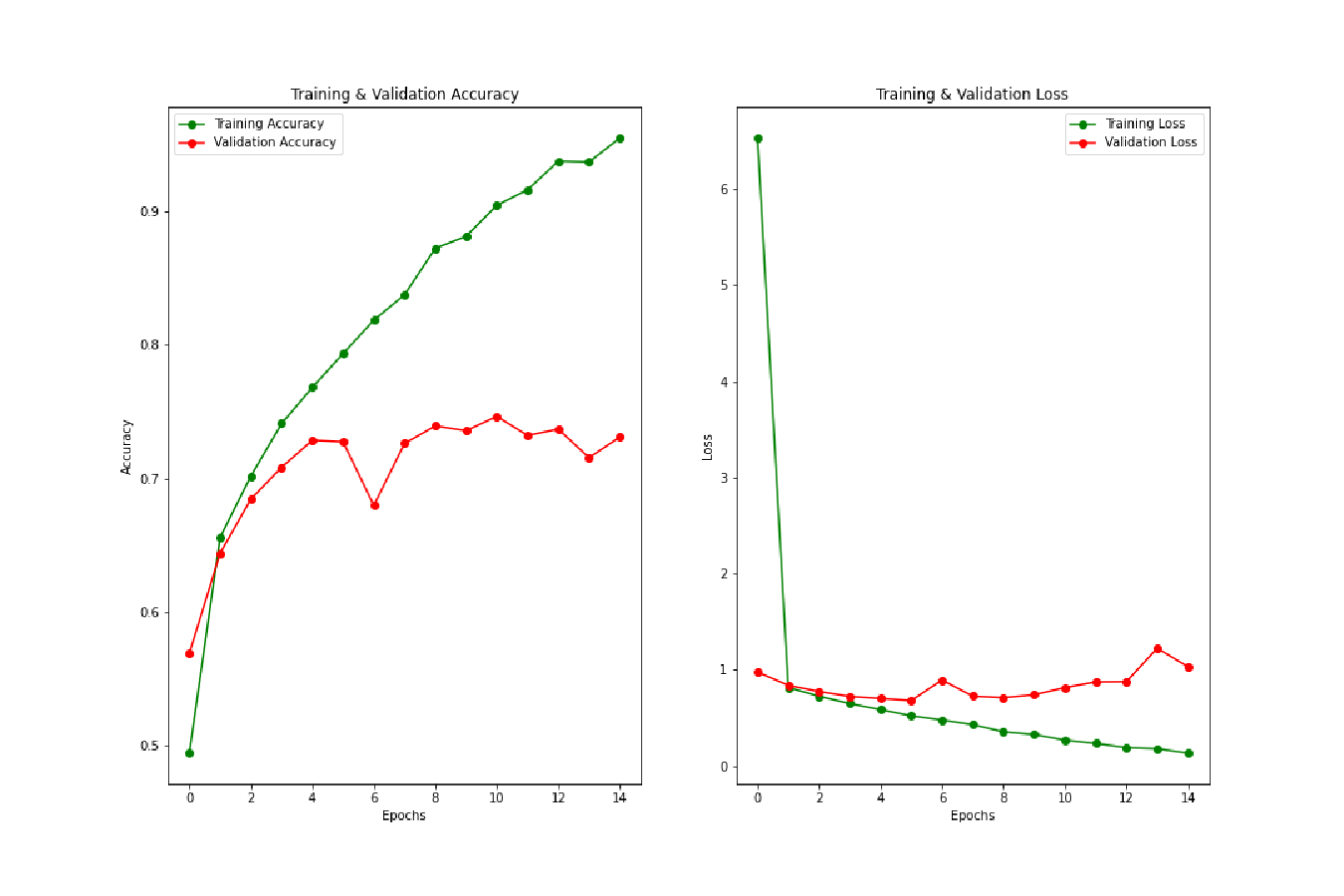
**7.RESULTS AND CONCLUSION**

* **WITHOUT IMAGE PROCESSING:**

* A training accuracy of **93.91%** accompanied by a validation accuracy of **68.33%** was achieved in the model trained on images without image processing.
* The training loss was **0.297.**
* The testing accuracy achieved by the same model was **70.78%.**
* The testing loss was **0.953.**
* The graphs for training/validation accuracies and training/validation losses across 15 epochs are showed in the following slide.



* **WITH IMAGE PROCESSING:**
* A training accuracy of **84.82%** accompanied by a validation accuracy of **72.86%** was achieved in the model trained on images without image processing.
* The training loss was **0.399**.
* The testing accuracy achieved by the same model was **74.72%**.
* The testing loss was **0.938**.
* The graphs for training/validation accuracies and training/validation losses across 15 epochs are shown in the following slide.



**8.INFERENCES**

* Applying Image Processing techniques improves the quality of the input images
* Improved quality provides better results
* The accuracy upon applying Image Processing techniques was boosted by almost 5%
* Hence, we can conclude that applying Image Processing techniques on our input images provides a beneficial boost to our models.

**9.CHALLENGES**

* The main challenge in conducting this experiment was the hardware.
* The original dataset consisted of 45000 images with 9 classes. However due to severe hardware constraints, such a large number of images could not be processed and trained.
* Therefore the dataset had to be brought down several times before settling for 6000 images of 3 classes.
* Despite reducing the dataset size certain Image Processing techniques such as edge detection were not applicable due to persistent hardware issues.
* Less number of images also led to overfitting of the model without image processing.

**10.FUTURE SCOPE AND IMPROVEMENTS**

* Better hardware availabilities can acutely boost the performance of the model by offering better computation speed and more processing power
* Larger datasets can be used to prevent overfitting and make a more generalized model for more classes
* More Image Processing techniques such as Denoising, Edge Detection, Transformations can be applied on the images along with better hyperparameter exploration such as that of Sharpening filters, Gamma values, etc.
* All of the points stated above can be combined to greatly increase the accuracy and performance of the model

**REFERENCES**

1.https://www.kaggle.com/datasets/andrewmvd/bone-marrow-cell-classification

2.https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=101941770