**Department of Electrical and Computer Engineering**

**North South University**



Directed Research (CSE498R)

**Monkeypox detection using Deep Learning and Explainable AI (XAI)**

|  |  |
| --- | --- |
| **Tashin Chowdhury** | **ID # 1911037642** |
| **Adiba Sharif** | **ID # 1912611642** |
| **Md. Shamsur Rahman Khan** | **ID # 1911843042** |

**Faculty Advisor**

**Dr. Mohammad Monirujjaman Khan**

**Associate Professor**

**ECE Department**

**Fall, 2022**

**DECLARATION**

This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

**Students’ name & Signature**

1. **Tashin Chowdhury**
2. **Md. Shamsur Rahman Khan**
3. **Adiba Sharif**

**APPROVAL**

The capstone project entitled “**Monkeypox Detection using Deep Learning and Explainable AI (XAI)**” by **Tashin Cowdhury (ID#1911037642), Md. Shamsur Rahman Khan (ID#1911843042),** and **Adiba Sharif (ID #1912611642)** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on May and has been accepted as satisfactory.

|  |
| --- |
|  |
| **Supervisor’s Signature** |
|  |
| **Dr. Mohammad Monirujjaman Khan  Associate Professor**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |
| **Department Chair’s Signature** |
| **Dr. Rajesh Palit**  **Professor & Chair**  Department of Electrical and Computer Engineering  North South University  Dhaka, Bangladesh. |

**ACKNOWLEDGMENT**

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report.

The capstone project program is very helpful to bridge the gap between the theoretical knowledge and real life experience as part of Bachelor of Science (BSc) program. This report has been designed to have a practical experience through the theoretical understanding.

We also acknowledge our profound sense of gratitude to all the teachers who have been instrumental for providing us the technical knowledge and moral support to complete the project with full understanding.

It is imperative to show our appreciation for our honorable faculty member Dr. Mohammad Monirujjaman Khan for his undivided attention and help to achieve this milestone. Also, our gratefulness is divine to the North South University, ECE department for providing us a course such as CSE 498R in which we could really work on this project and materialize it the way we have dreamt of.

We thank our friends and family for their moral support to carve out this project and always offer their support.

**ABSTRACT**

The monkeypox virus (MPXV), a member of the orthopoxviral genus, is the infectious illness responsible for monkeypox. It was discovered in a monkey for the first time in 1959 at a research institute. That’s why it is known as the monkeypox virus in Denmark. Later, the first human case was identified in the Republic of Congo in 1970 after a young kid with symptoms similar to smallpox was admitted to the hospital. It spreads to people by coming into close contact with them or contaminated objects. Primarily it has appeared in Africa, but recently it has spread to more than 50 nations, with 3,413 confirmed cases and one death. There are currently just two recognized varieties of the monkeypox virus: one, the Central Africa clade, and another, the West Africa clade. Our aim is to build a model which can predict whether the skin decease a patient is suffering from is the contagious monkeypox or not with a high level of accuracy. The model we build is not only will help to detect whether it's monkey pox or not; but also, it will show possible region that might cause monkeypox which will increase the reliability of our model in the medical field & another important motive behind our model is make it user-friendly and marketable in the context of Bangladesh medical services. To train our multiple models for accurate prediction, we take various approaches, such as after setting the parameters & batch size we created some CNN layers and for better accuracy we set epoch as 50, applied AI tool like Lime , explainable masking method for image classifiers and heatmap data visualization. After training our model gives us almost 100% accurate results, which is a remarkable achievement as no other approach taken towards this issue has given results with this much accuracy. Overall, our model can have a positive impact on our society as well as bring revolutionary changes to medical science by generating output at this satisfactory level. Our model can create the smallest possible mask that changes the model’s original prediction on an image when given a perturbed image (that same image with the mask applied). And, by implementing heatmap data visualization, we can view and process visuals way more efficiently than written text or written data.

**Table of Content**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | Page |
|  | **Chapter1: Introduction**………………………………………………. | | 1 |
| 1.1 | Introduction………………………………………………………………………… | | 2 |
| 1.2 | Project Details……………………………………………………………………… | | 2 |
| 1.3 | Project Goals……………………………………………………………………… | | 2 |
| 1.4 | Summary…………………………………………………………………………… | | 2 |
|  |  | |  |
|  | **Chapter 2: Motivation**………………………………………………. | | 3 |
| 2.1 | Introduction………………………………………………………………………… | | 4 |
| 2.2 | Motivation towards our project……………………………………………………. | | 4 |
| 2.3 | Summary…………………………………………………………………………… | | 4 |
|  |  | |  |
|  | **Chapter 3: Related Works**…………………………………………... | | 5 |
| 3.1 | Introduction………………………………………………………………………… | | 6 |
| 3.2 | Systems related to our project……………………………………………………… | | 6 |
| 3.3 | Problems with the current system………………………………………………….. | | 9 |
| 3.4 | Proposed Solution………………………………………………………………...... | | 10 |
| 3.5 | Summary………………………………………………………………………….. | | 10 |
|  |  | |  |
|  | **Chapter 4: Technical Design**…….…………………………………... | | 11 |
| 4.1 | Introduction………………………………………………………………………… | | 12 |
| 4.2 | Technical Design: System Level…………………………………………………… | | 12 |
| 4.3 | Summary…………………………………………………………………………… | | 13 |
|  |  | |  |
|  | **Chapter 6: Methodology**……..………………………………………. | | 14 |
| 5.1 | Introduction……………………………………………………………… | | 15 |
| 5.2 | Software Design: Module Level…………………………………………… | | 15 |
| 5.3 | Summary…………………………………………………………………………… | | 24 |
|  |  | |  |
|  | **Chapter 6: Design Results and Analysis** ………………..………… | | 25 |
| 6.1 | Introduction……………………………………………………………… | | 26 |
| 6.2 | Software Design: Module level …………………………………………………… | | 26 |
| 6.3 | Summary………………………………………………………………………….. | | 33 |
|  |  | |  |
|  | **Chapter 7: Skills**……………………………………………………... | | 35 |
| 7.1 | Introduction……………………………………………………………… | | 36 |
| 7.2 | Skills obtained……………………………………………………………………... | | 36 |
| 7.3 | Summary………………………………………………………………… | | 37 |
|  |  | |  |
|  | **Chapter 8: Future Work**………………….………………………….. | | 38 |
| 8.1 | Introduction………………………………………………………………………… | | 39 |
| 8.2 | Future scope of work………………………………………………………………. | | 39 |
| 8.3 | Summary…………………………………………………………………………… | | 39 |
|  | |
|  | |
|  |  | |  |
|  | **Chapter 9: Design Impact**………………………….………………… | | 40 |
| 9.1 | Introduction………………………………………………………………………… | | 41 |
| 9.2 | Environmental Impact……………………………………………………………… | | 41 |
| 9.3 | Economic Impact…………………………………………………………………… | | 41 |
| 9.4 | Social Impact……………………………………………………………………….. | | 41 |
| 9.5 | Sustainability…………………………………………………..…………………… | | 41 |
| 9.6 | Summary…………………………………………………………………………… | | 42 |
|  |  | |  |
|  | **Chapter 10: Compliance with Standards**………….………………... | | 43 |
| 10.1 | Introduction………………………………………………………………………… | | 44 |
| 10.2 | Compliance with IEEE standard…………………………………………………… | | 44 |
| 10.3 | Compliance with US standard……………………………………………………… | | 44 |
| 10.4 | Summary…………………………………………………………………………… | | 45 |
|  |  | |  |
|  | **Chapter 11: Conclusion**……………………………...……………… | | 46 |
|  |  | |  |
|  | **Bibliography**…………………………………………………………... | | 48 |
|  | **Appendix: Software Listing**………………………………..………… | | 51 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure No. | Figure caption | Page No. |
| 4.1 | Data Flow Diagram | 12 |
| 5.1 | Flow Chart of our proposed model | 15 |
| 5.2(a) | Others | 16 |
| 5.2(b) | Monkey Pox | 16 |
| 5.3 | Intuition of Adam Optimizer | 17 |
| 5.4 | Software Design of CNN | 18 |
| 5.5 | Explanation of the functioning of the complete system and all subsystems | 22 |
| 6.1 | Classification Report of the entire model | 26 |
| 6.2 | Accuracy and Loss Graph | 27 |
| 6.3 | Confusion Matrix | 28 |
| 6.4 | Deep Learning as a Black-Box | 28 |
| 6.5(a) | Prediction of Monkeypox. | 30 |
| 6.5(b) | Lime Image Interpretation with Bounded Regions. | 31 |
| 6.5(c) | Lime Image Interpretation with Boundaries Overlapping each other strength wise. | 32 |
| 6.5(d) | Lime Overall Bounded Regions for Interpretation. | 32 |
| 6.5(e) | Lime Heat map of the Predicted Image | 33 |

**CHAPTER 1**

**Introduction**

**1.1 Introduction**

The infectious sickness that causes monkeypox is caused by the monkeypox virus (MPXV), a member of the orthopoxviral genus. It was initially detected in a monkey at a scientific center in 1959 [1]. In Denmark, it is known as the monkeypox virus. Later, in 1970 [2], a little child with symptoms identical to smallpox was taken to a hospital in the Republic of Congo, and the first human case was recognized. It spreads to humans by intimate contact with them or infected things [3]. It first started in Africa, but it has lately spread to more than 50 countries, with 3,413 confirmed cases and one fatality. There are now only two identified types of the monkeypox virus: one from Central Africa and one from West Africa [4].

**1.2 Project Details**

To train our multiple models for accurate prediction, we take various approaches, such as after setting the parameters & batch size we created some CNN layers and for better accuracy we set epoch as 50, applied AI tool like Lime , explainable masking method for image classifiers and heatmap data visualization. After training Deep feature extraction has been carried out using deep CNN models that have already been trained, including ResNet50, InceptionV3, MobileNetV2, and VGG16. The accuracy of the classifications was used as the performance indicator in this study. The pre-trained tailored models, such as MobileNetV2, InceptionV3, and VGG16, each achieved 98% accuracy while the developed model achieved 93% accuracy and 98% validation accuracy. However, our model provides findings that are nearly 100% accurate, which is a remarkable accomplishment given that no other solution to this problem has produced results with this level of accuracy.

**1.3 Project Goals**

There are two major goals that we want to focus on with our project. They are as follows:

i) Explain the observations using Explainable AI: Explainable AI (XAI) is a collection of tools and frameworks that are natively linked with several Google products and services to assist in comprehending and interpreting predictions provided by machine learning models. With it, we can debug models, enhance their performance, and aid in the behavioral understanding of others. To do so, we have used the tools LIME to interpret our black box model.

ii) Explain the behavior of the model using Masking method & Heatmap visualization: Our method is a type of local explanation, which seeks to explain how a model behaves given a particular input. Our method’s objective is to create the smallest possible mask that changes the model’s original prediction on an image when given a perturbed image. And, heatmap visualization or heatmap data visualization is a method of graphically representing numerical data where the value of each data point is indicated using colors.

**1.4 Summary**

Overall, our model can have a positive impact on our society as well as bring revolutionary changes to medical science by generating output at this satisfactory level. Our model can create the smallest possible mask that changes the model’s original prediction on an image when given a perturbed image (that same image with the mask applied). And, by implementing heatmap data visualization, we can view and process visuals way more efficiently than written text or written data.

**CHAPTER 2**

**Motivation**

**2.1 Introduction**

Our aim is to build a model which can predict whether the skin decease a patient is suffering from is the contagious monkeypox or not with a high level of accuracy. The model we build is not only will help to detect whether it's monkey pox or not; but also, it will show possible region that might cause monkeypox which will increase the reliability of our model in the medical field & another important motive behind our model is make it user-friendly and marketable in the context of Bangladesh medical services.

**2.2 Motivation towards our project**

Since all previous studies focused on the effects of monkeypox or monkeypox detection using different machine learning or deep learning algorithms, we attempted to highlight the difference we can see in images of monkeypox and other skin diseases that are more prominent features by implementing an interpretable model that explains the factor using xAI.

**2.3 Summary**

Overall, by producing output at this reasonable level, our approach may have a good influence on our society as well as bring revolutionary advances to medical research. When given any picture as input, our model can generate the smallest feasible mask that affects the model's initial forecast (that same image with the mask applied) and by using heatmap visualization, we view and process visuals way more efficiently than written text or written data.

**CHAPTER 3**

**Related Works**

**3.1 Introduction**

The monkeypox virus (MPXV), a member of the orthopoxviral genus, is the infectious illness responsible for monkeypox. It was discovered in a monkey for the first time in 1959 at a research institute. That’s why it is known as the monkeypox virus in Denmark. Later, the first human case was identified in the Republic of Congo in 1970 after a young kid with symptoms similar to smallpox was admitted to the hospital. It spreads to people by coming into close contact with them or contaminated objects. Primarily it has appeared in Africa, but recently it has spread to more than 50 nations, with 3,413 confirmed cases and one death. There are currently just two recognized varieties of the monkeypox virus: one, the Central Africa clade, and another, the West Africa clade.

**3.2 Systems related to our project**

Islam, T.; Hussain, M.A.; Chowdhury, F.U.H.; Islam, mentioned in their paper about, B.M.R. AWeb-scraped Skin Image Database of Monkeypox, Chickenpox, Smallpox, Cowpox, and Measles, that they used an image dataset of monkey pox made by Hayward, G. [6] with the help of web scraping, in the dataset there was a comprehensive picture collection of both sick and healthy skin for anybody can found and used. The images of skins show symptoms of cowpox, measles, chickenpox, monkeypox and smallpox. Then, from the pictures of measles, chickenpox and monkeypox the authors creates the Monkeypox Skin Lesion Dataset (MSLD). The dataset is publicly accessible. Initial approaches included a 3-fold cross-validation experiment and increasing the sample size with more data. The second stage involved categorizing ailments using pre-trained deep learning models including VGG-16, ResNet50, and InceptionV3 (e.g., monkeypox). ResNet50 achieved the highest level of overall accuracy. The authors proposed a deep learning model for identifying monkeypox disease based on picture data collection and implementation, using a modified version of VGG16. Given that the dataset was created by gathering photographs from many open-source publications and websites, it is safer to utilize and disseminate such data for creating and deploying any machine-learning model. The VGG16 model with the modifications was utilized in two different research. According to the results of both trials, this model may successfully identify patients who have monkeypox. The model's capacity to anticipate and extract such properties enables the development of greater understanding of the characteristics of the monkeypox virus. [7]

Chiranjibi Sitaula and Tej Bahadur Shahi mentioned in their paper about Monkeypox Virus Detection Using Pre trained Deep Learning based Approaches that, they compare 13 different pre-trained deep learning (DL) models for monkeypox detection. In order to do this, they first refine the dataset by adding common custom layers for all of them. Then, they analyze the findings using four well-known metrics: Precision, Recall, F1-score, and Accuracy. They ensembled the top-performing DL models after identifying which ones performed the best by utilizing a majority vote over the probabilistic outputs they produced. They conducted their trials on a publically accessible dataset, and their suggested ensemble strategy produced average Precision, Recall, F1-score, and Accuracy values of 85.44%, 85.47%, 85.40%, and 87.13%, respectively. The model's capacity to predict and extract such traits led to these optimistic results, which outperformed the state-of-the-art approaches, suggesting that the proposed methodology is applicable to health practitioners for mass screening. [8]

Additionally, Glock et al. used a transfer learning strategy to detect the measles disease. Using the ResNet-50 model on the diverse rash image dataset, they obtained a sensitivity of 81.7%, specificity of 97.1%, and accuracy of 95.2% [9]. In addition, a big-data strategy using ensemble learning was suggested for detecting the Ebola virus sickness. Using the Apache Spark and Kafka frameworks, they used genetic algorithms (GA) and artificial neural networks (ANN) to extract knowledge from large amounts of data. More recently, Ahsan et al. used web mining techniques to gather images for the categories of monkeypox, chickenpox, measles, and normal, which were then validated by specialists. Later, they used the VGG-16 model to evaluate a transfer learning strategy while taking into account two methods. When identifying monkeypox without the use of augmented data, they reported an accuracy of (97%); however, the accuracy fell to (78%) when the augmentation was used. [10]

A few studies have also employed deep learning models to detect other diseases including chicken pox, herpes, and so forth in addition to the COVID-19 virus. For instance, using deep learning (DL)-based methods, Sandeep et al.[11] explored the detection of several skin conditions including psoriasis, chicken pox, vitiligo, melanoma, ringworm, acne, lupus, and herpes. They developed a Convolutional Neural Network (CNN) to classify the skin lesion into eight disease classes and compared their solution with the help of the VGG-16 pre-trained model[12]. Their technique provided a 78% detection accuracy. It was suggested in to use CNN for low-cost image analysis for the detection of the Herpes Zoster Virus (HZV). When evaluated on 1,000 photos, the early detection of HZV produced an accuracy of 89.6%. [13]

Md Manjurul Ahsan, Muhammad Ramiz Uddin, Mithila Farjana, Ahmed Nazmus Sakib, Khondhaker Al Momin and Shahana Akter Luna in their paper about Image data collection and implementation of deep learning-based model in detecting monkeypox disease using modified VGG16 mentioned that, as there were no public dataset available there they developed a dataset “Monkeypox 2022”, it can be found in github repository. The dataset was developed by gathering photos from a variety of open-source and internet resources that do not place any limitations on use, even for commercial purposes, providing a safer approach to utilize and disseminate such data for developing and deploying any kind of ML models. Additionally, they created and assessed a modified VGG16 model that consists of two separate studies, Study One and Study Two. Their preliminary computational findings show that our proposed model can correctly identify monkeypox patients in Studies One and Two with an accuracy of 97.2% (AUC = 97.2) and 88.88% (AUC = 0.867), respectively. Additionally, they provide a deeper understanding of specific aspects that define the start of the monkeypox virus by describing how their model predicts and extracts features using Local Interpretable Model-Agnostic Explanations (LIME). [14]

Shams Nafisa Ali, Md. Tazuddin Ahmed, Joydip Paul, TasnimJahan, S. M. SakeefSani, Nawsabah Noor, Taufiq Hasan in their paper about Monkeypox Skin Lesion Detection Using Deep Learning Models: A Feasibility Study mentioned that as the dataset of monkeypox was not available so they develop the ``Monkeypox Skin Lesion Dataset (MSLD)" consisting skin lesion images of monkeypox, chickenpox, and measles. A 3-fold cross-validation experiment is set up, and the sample size is increased through data augmentation. The second phase is categorizing diseases like monkeypox using a number of pre-trained deep learning models, including VGG-16, ResNet50, and InceptionV3. Also created is an ensemble of the three models. The ensemble system and VGG16 both reached accuracy levels of 79.26(1.05%) and 81.48(6.87%), respectively. ResNet50 has the highest overall accuracy of 82.96(4.57%). [15]

Abdelhamid, A.A.; El-Kenawy, E.-S.M.; Khodadad, N.; Mirjalili, S.; Khafaga, D.S.; Alharbi, A.H.; Ibrahim, A.;Eid, M.M.; Saber mentioned in their paper about M. Classification of Monkeypox Images Based on Transfer Learning and the Al-Biruni Earth Radius Optimization Algorithm, that In image-based diagnostics, such as cancer detection, tumor cell identification, and COVID-19 patient detection, machine learning has recently showed excellent promise. Therefore, as monkeypox spreads to human skin, a similar technique might be used to detect it. Two strategies are suggested in this research to improve the classiﬁcation accuracy of monkeypox images. The suggested approaches are based on feature extraction by transfer learning, feature selection via meta-heuristic optimization, and parameter optimization for multilayer neural networks. The Al-Biruni Earth radius algorithm, the sine cosine algorithm, and the particle swarm optimization technique are used as meta-heuristic optimization algorithms. The GoogleNet deep network is used for feature extraction. A new binary hybrid algorithm and a new hybrid method for optimizing the neural network's parameters are suggested based on these algorithms. A publicly accessible dataset is used to assess the proposed algorithms. Ten evaluation criteria were used to analyze the proposed optimization of feature selection for monkeypox classification. A set of statistical tests were also run to examine the effectiveness, significance, and robustness of the suggested algorithms. The obtained findings attest to the proposed approaches' superiority and efficacy over alternative optimization techniques. 98.8% of classifications were correctly made on average. [16]

**3.3Problems with the current systems**

Our aim is to build a model which can predict whether the skin decease a patient is suffering from is the contagious monkeypox or not with a high level of accuracy. The model we build is not only will help to detect whether it's monkey pox or not; but also, it will show possible region that might cause monkeypox which will increase the reliability of our model in the medical field & another important motive behind our model is make it user-friendly and marketable in the context of Bangladesh medical services.

**3.4 Proposed Solution**

We implemented these approaches to solve the problems we are facing in our current systems

i) Implemented Explainable AI: Explainable AI (XAI) is a collection of tools and frameworks that are natively linked with several Google products and services to assist in comprehending and interpreting predictions provided by machine learning models. With it, we can debug models, enhance their performance, and aid in the behavioral understanding of others. To do so, we have used the tools LIME to interpret our black box model.

ii) Implemented Masking method & Heatmap visualization: Our method is a type of local explanation, which seeks to explain how a model behaves given a particular input. Our method’s objective is to create the smallest possible mask that changes the model’s original prediction on an image when given a perturbed image. And, heatmap visualization or heatmap data visualization is a method of graphically representing numerical data where the value of each data point is indicated using colors.

**3.5 Summary**

By producing output at this satisfying level, our model can have a positive impact on society and lead to revolutionary improvements in medical science. When given a disturbed image, our model can produce the smallest mask possible that modifies the model's original prediction on the image (that same image with the mask applied). Additionally, we can examine and analyze images much more quickly than written text or written data by utilizing heatmap data visualization.

**CHAPTER 4**

**TECHNICAL DESIGN**

**4.1 Introduction**

We go over the technical design of our system in this chapter. It would be simpler to conceptualize the entire system's data flow by going through the system level design.

**4.2 Technical Design: System Level**

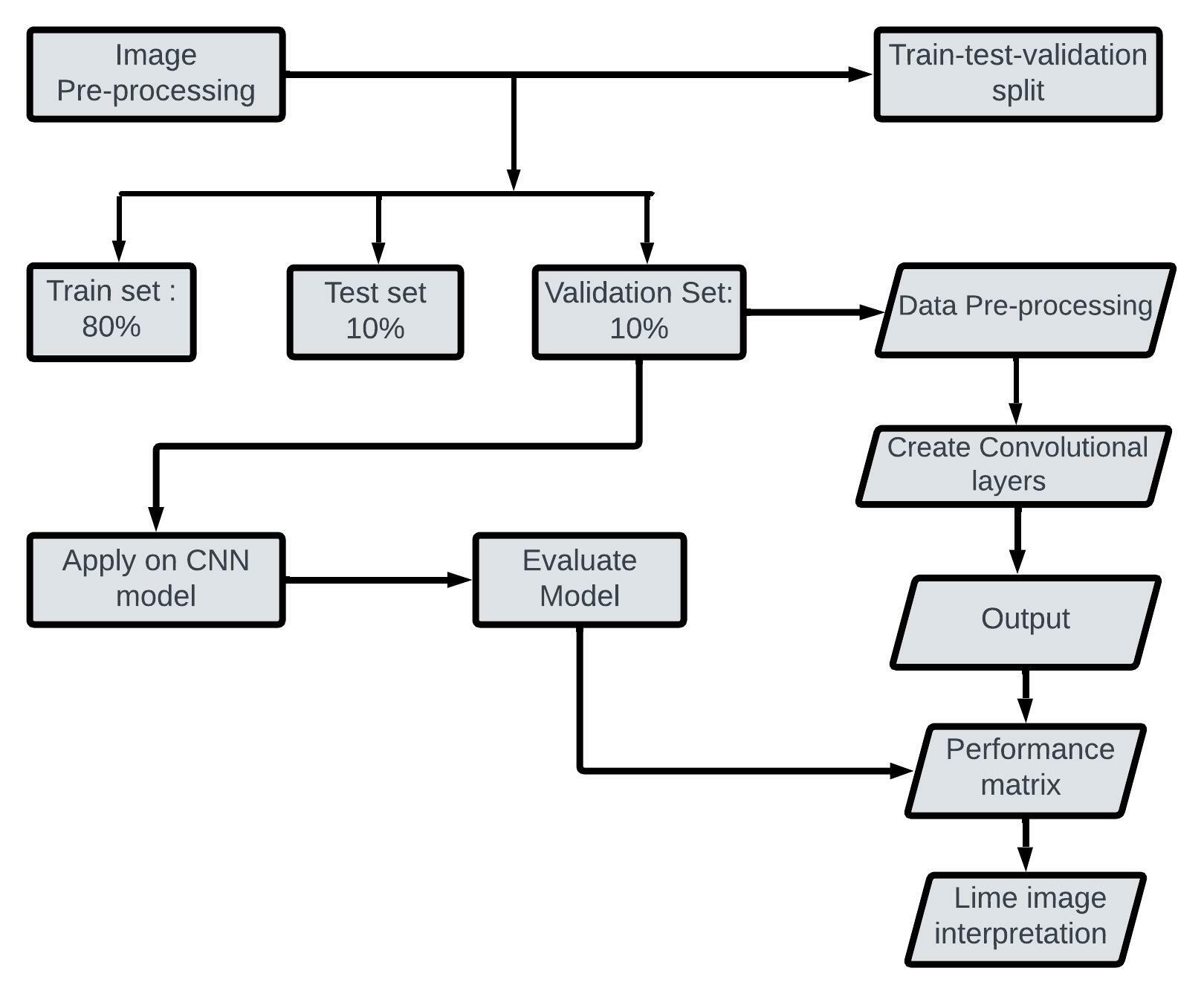


Figure 4.1: Data Flow Diagram

The dataset was obtained from the open-source Kaggle. It is an image dataset. The dataset consists of 228 files belonging to 2 classes; which are 'Monkeypox\_augmented', 'Others\_augmented'. Here, 'Others\_augmented' has the images of different skin disease rather than monkeypox. Then, as the data cleaning concluded, we split the data into 80% for training and 20% for testing.

**4.3 Summary**

we have implemented a neural network to predict monkeypox and then explained the traditional black box model using explainable AI. We have also used traditional machine learning mechanisms to interpret. We used XAI on our image dataset. And, in our deep learning model, we used masking method; which aim is to describe how a model performs in response to a certain input based on what parts of the image did the model most rely on to make its prediction and how much did this model rely on those features to make its prediction. And, lastly, we implemented heatmap visualization; which is a graphical way to visualize visitor behavior data in the form of hot and cold spots employing a warm-to-cool color scheme.

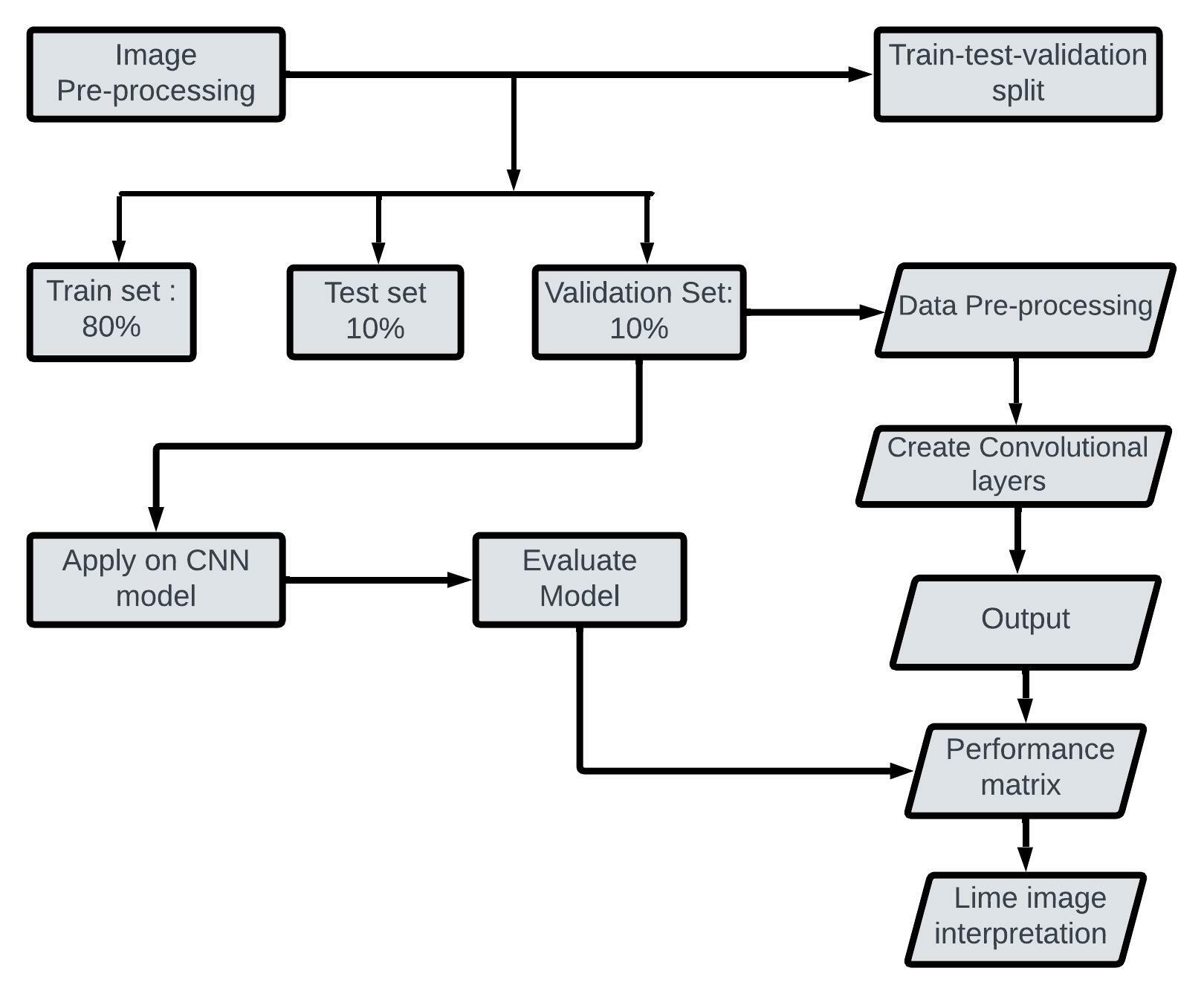
**CHAPTER 5**

**METHODOLOGY**

**5.1 Introduction**

We have implemented a neural network to predict monkeypox and then explained the traditional black box model using explainable AI. We have also used traditional machine learning mechanisms to interpret. We used XAI on our image dataset. And, in our deep learning model, we used masking method; which aim is to describe how a model performs in response to a certain input based on what parts of the image did the model most rely on to make its prediction and how much did this model rely on those features to make its prediction. And, lastly, we implemented heatmap visualization; which is a graphical way to visualize visitor behavior data in the form of hot and cold spots employing a warm-to-cool color scheme. In this section, we are going to discuss our approach in detail.

**5.2 System Architecture:**

****

**Figure 5.1:** Flow chart of our proposed method.

***Materials and Tool:***

Programming with Python is the best choice for data analysis. Python programming is very effective with deep learning-based difficulties due to Python's extensive library access. Large datasets and model training were handled online using Google Colab, Anaconda Navigator, and Jupyter Notebook to take advantage of a personal GPU for dataset preprocessing

***Data Preprocessing:***

The dataset was obtained from the open-source Kaggle. It is an image dataset. The dataset consists of 228 files belonging to 2 classes; which are 'Monkeypox\_augmented', 'Others\_augmented'. Here, 'Others\_augmented' has the images of different skin disease rather than monkeypox. Then, as the data cleaning concluded, we split the data into 80% for training and 20% for testing.



Figure 5.2 (a) Figure 5.2(b)

Here are some examples of our data from the dataset; where, figure (a) shows not monkeypox and figure (b) shows monkeypox

***Data Augmentation:***

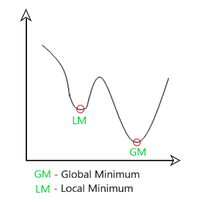
Data Augmentation The training and validation sets underwent augmentation. The image count of the training and validation set went from 228 to 3192 after data augmentation.

***Transfer Learning:***

Finally, we have used transfer learning models to train our model. As our dataset is split into an 80:20 ratio for training and testing, we then created 5 convolutional layers after setting some parameters; as we set batch size 64 for the image size 256 & to train the CNN model we set epoch 50. Then, in order to implement deep learning, we used CNN to train the model and determine the accuracy and F1 score.In our model Total parameter is 183,682; Where trainable parameters: 183,682(so, no non-trainable parameter exists). Then we did a comparative analysis based on performance metrics such as accuracy, precision, recall, F1-score.

***Optimizer (adam):***

Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments; which could be implemented in various deep learning applications such as computer vision and natural language processing in the future years [17].



**Figure 5.3:** Intuition of Adam Optimizer.

Here, we regulate the gradient descent rate to reach the global minimum with the least amount of oscillation possible while passing the local minima obstacles with sufficiently large steps (step-size). Thus, combining the advantages of the aforementioned approaches will effectively obtain the global minimum. The results of the Adam optimizer are generally better than every other optimization algorithm, have faster computation time, and require fewer parameters for tuning. Because of all that, Adam is recommended as the default optimizer for most of the applications.

***System Architecture:***

An overview of the complete system is provided by the system architecture. In this architecture, an image of monkeypox or another skin condition is used as the input, and the output is an image prediction. It will then predict whether the image depicts monkeypox or not. There are three channels and a 256 x 256 input form. The filter size is 32 with padding, the kernel size is 3, and the activation function is Rectified Linear Unit (ReLU) in the first two layers of the designed architecture.

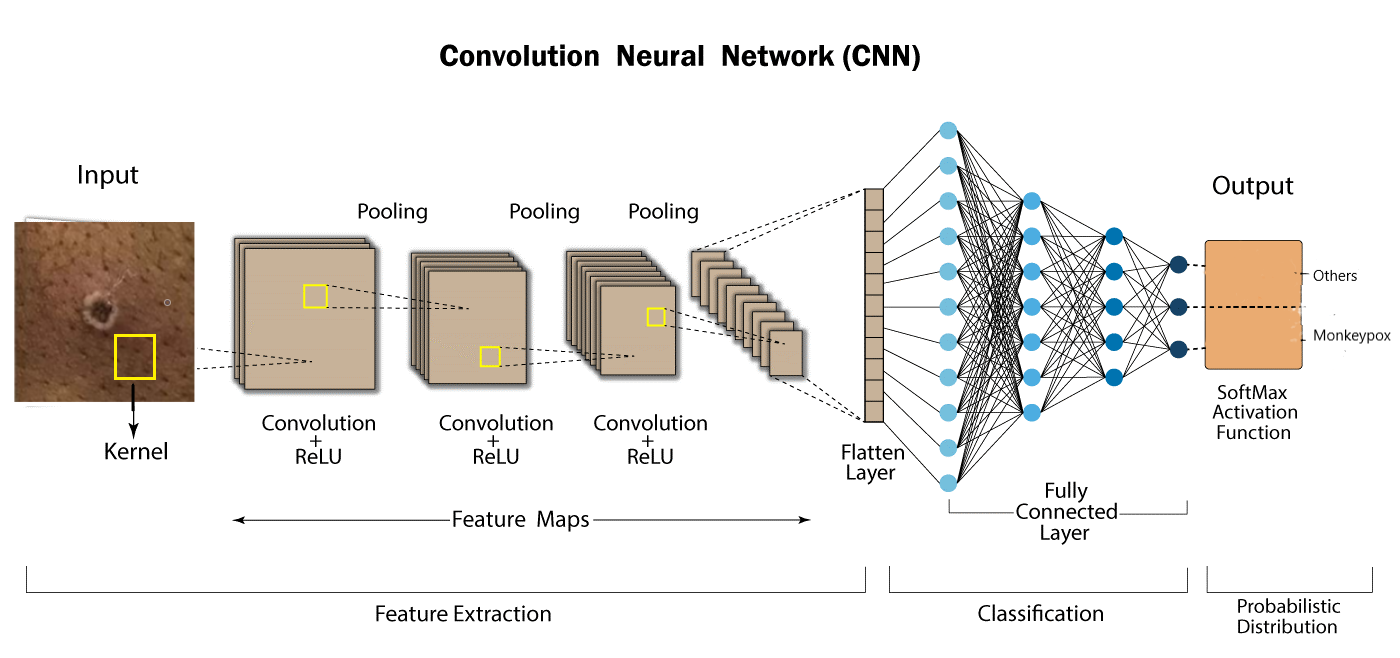


Figure 5.4: Software Design

**Convolutional Layer:**

The basic layer of CNN is the convolutional layer. This is in charge of deciding the design characteristics. In this layer, a filter is applied to the input image. The output of the same filters is used to generate the function map through a convolution operation.

Using a convolution operation, sets of weights are multiplied with the input. A filter is made up of an array of input data multiplied by a two-dimensional collection of weights. When a filter-sized patch of the input and the filter are multiplied together, it is known as a dot product and yields a single value. Between the input's filter-sized patch and the filter, this product is applied. The same filter is used to multiply the input from several places, and it is smaller than the input. The filter is created as a unique method to recognize particular elements as it meticulously goes through the entire image.

Assume that the NN input is V∈R^(A×B), where A is the total number of input frequency bands and B is the number of features that represent each input frequency band. In the case of filter bank features, B denotes the size of the filter bank function vector. Assume that v = [v1 v2... vB], with vB standing for the band b function vector. The activations of the convolution layer can be calculated as

where h\_(j,k) is the j^th feature map’s convolution layer output of the convolution layer band of k^th, s indicates the filter scale, w\_(b,j) indicates the weight vector for the j^th filter’s bth band, a\_j is the j^th feature map's bias, and (x) represents the activation function. [18]Lastly, Report Generator will collect all test results against the patient’s ID and generate a final report to the system and will notify that all tasks have been performed for the specific patient. Admin will observe all tasks in the system and will create the users when required. Admin has the full privilege to make any changes to the system.

**Pooling Layer**

By enabling the down sampling of features, the pooling layer summarizes the existence of features. It has some spatial invariance and is often applied following a convolution layer. The average presence of a function and the most activated presence of a function are summarized by the two widely used pooling techniques, average pooling and max pooling. [19].

In fact, the pooling layer rids the images of superfluous details and transforms them into literate images. When using average pooling, the layer consistently averages the value of the current view. When utilizing maxpooling, the layer repeatedly chooses the highest value from the filter's active view. The max-pooling strategy picks only the maximum value using the matrix size set in each feature map, resulting in fewer output neurons. As a result, the image shrinks dramatically yet the situation stays the same. A dropout layer is utilized to avoid overfitting, and a pooling layer is crucial for minimizing the amount of feature mappings and network parameters.

The activation of max pooling can be calculated as follows:

(2)

where p\_(j,m) is the performance of the pooling layer of the jth function map and the mth pooling layer band, n is the subsampling factor, r is the pooling scale which is the number of bands to be pooled together, and n is the subsampling factor.

**Flatten Layer :**

The flattened layer is used to construct a single one-dimensional feature that is both long and thin, as well as to transform data from the matrix into a one-dimensional array for usage in the fully linked layer. Vectors can be flattened, as an option. The single vector is then linked to the final classification model, also referred to as a fully connected layer [20]. A single set of fully connected layers connects all of the pixel data. The final steps of CNN are layer flattening and layer connectivity. It is transformed into a one-dimensional array in order to be ready for the following fully linked layer of photo categorization.

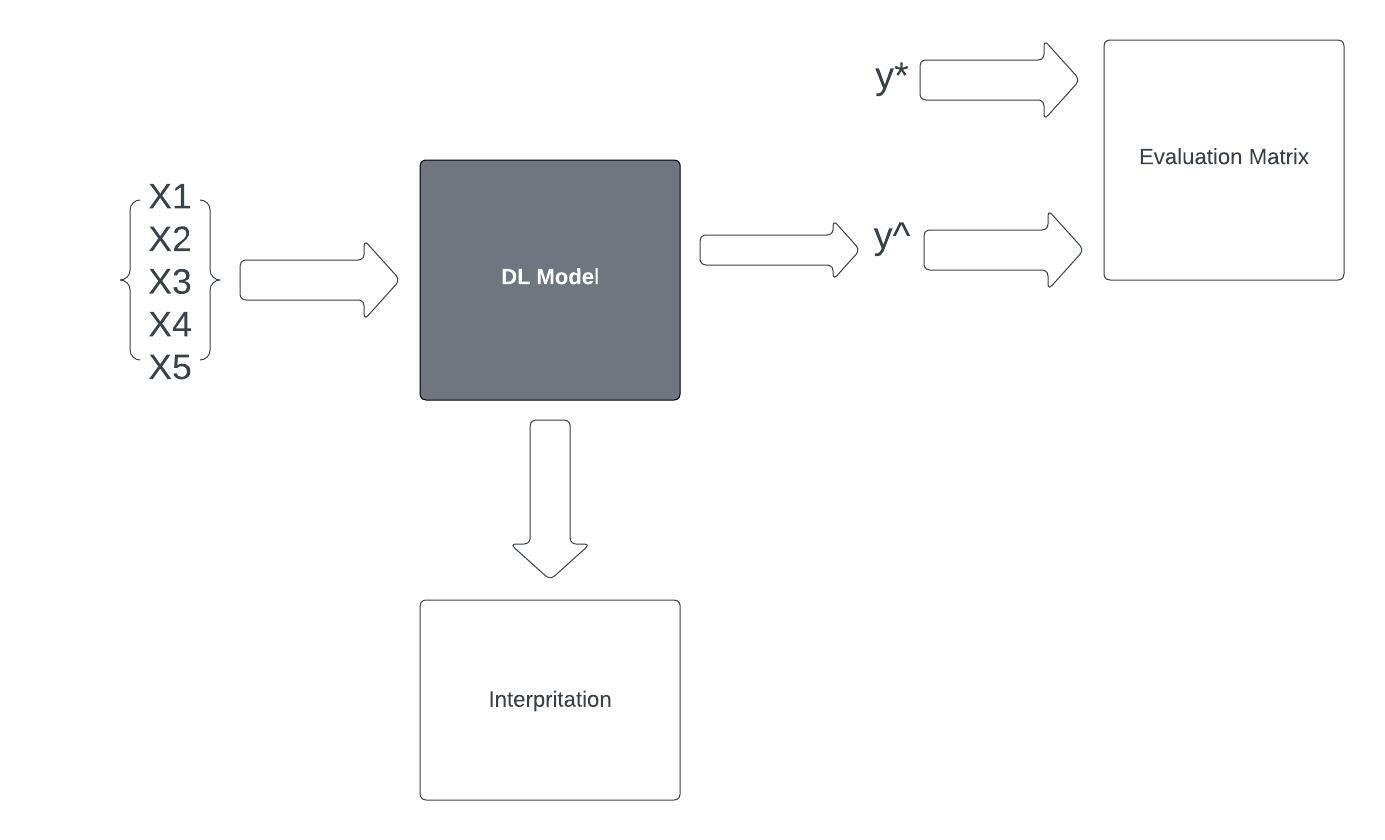
**Fully Connected Layer:**

Fully linked layers, on which CNNs mostly rely, have shown to be very beneficial for computer vision picture recognition and classification. The initial stages of the CNN method, which divides the image into characteristics and examines each separately, are convolution and pooling. [21].

Each input is flattened and connected to every neuron in a fully connected layer. A fully connected layer that is frequently employed is the Rectified Linear Unit (ReLU) activation function. In the last layer of the fully connected layer, the output images were predicted using the softmax activation function. A fully connected layer is used in the architecture of the convolutional neural network. These are the convolutional neural network's final and most crucial layers.

Explainable AI (LIME):

Explainable AI (XAI) is a collection of tools and frameworks that are natively linked with several Google products and services to assist you in comprehending and interpreting predictions provided by your machine learning models. With it, you can debug models, enhance their performance, and aid in the behavioral understanding of others. To do so, we interpreted our black box model using the tools Local Interpretable Model-agnostic Explanations (LIME). It is one of the most widely used Explainable AI (XAI) techniques for describing how deep learning and machine learning models function. LIME can solve regression and classification problems with model-independent local explanations, and it works with both structured and unstructured datasets, including text and image data. LIME interpreted a single output and explained that the trained factor is responsible for the specific prediction. In other words, machine learning algorithms' output and outcomes can now be understood and trusted by human users thanks to a set of procedures and techniques known as explainable artificial intelligence (XAI)[22]. An AI model, its anticipated effects, and potential biases are all described in terms of explainable AI. It contributes to defining model correctness, fairness, transparency, and outcomes in decision-making supported by AI[23]. A business must establish trust and confidence when putting AI models into production. A company can adopt a responsible approach to AI development with the aid of AI's explanation abilities. Humans find it difficult to understand and trace the steps taken by the algorithm as AI develops. The entire calculating procedure is transformed into what is known as a "black box," which is difficult to understand. These black box models are created using data. Furthermore, nobody, not even the engineers or data scientists who developed the algorithm, can comprehend or describe what exactly is going on inside of them, let alone how the AI algorithm came to a particular conclusion. Understanding how an AI-enabled system produced a particular result has several benefits. Explainability can help developers ensure that the system is working as it should, may be required to meet regulatory requirements, or may be critical in allowing people affected by a decision to contest or change the outcome. [24]

****

**Figure 5.5:** Explanation of the functioning of the complete system and all subsystems

Using the traditional approaches of feature importance and other frameworks like LIME and SHAP, explaining tabular datasets is still simple. Explaining sophisticated deep learning models that have been trained on unstructured data, like photographs, is, nevertheless, always a challenge. Since deep learning models have the potential to perform automatic feature extraction, they are typically far more effective than traditional ML models on picture data. Complex low-level elements like stripes, edges, curves, corners, and patterns can be extracted, as well as higher-level features like more substantial shapes and specific areas of the item. Since they are groups of pixels from the image that span a specific area of the image, these higher-level features are frequently referred to as super pixels or Regions of Interest (RoI) in the image. The high-level features, however, can be understood by anyone who is not technically inclined because they are related to the photos in terms of the higher-level features, whereas the low-level features are not. Similar principles apply to LIME's operation. The algorithm looks for super pixels in images that influence the model's judgment in a favorable or negative way. Let's examine LIME's application to the explanation of image classifiers.

**Explainable Masking Method for Image Classifiers:**

By implementing mask method, a model is able to create the smallest possible mask that changes the model’s original prediction on an image when given a perturbed image (that same image with the mask applied). The area which the mask affects is blurred. By minimizing the regions that are masked, we can identify the most essential features of the image according to the model and quantify how much those regions affect the model’s prediction by measuring the change in the model’s confidence on the original label before and after the mask is applied. As most users of these models are not ML researchers nor engineers. They most likely cannot modify the model in any way to interpret if its decisions align with the desired outcomes of the users. We therefore use mask method that are easily applied on black-box models, models which we cannot modify nor know its internal components, while being intuitive and accessible to both the expert and to the layperson

**Heatmap Data Visualization:**

Heatmap visualization or heatmap data visualization is a method of graphically representing numerical data where the value of each data point is indicated using colors. The most commonly used color scheme used in heatmap visualization is the warm-to-cool color scheme, with the warm colors representing high-value data points and the cool colors representing low-value data points. The reason that visualization of data through methods like heatmap has become so popular is that humans, in essence, are visual beings. Many studies on human psychology and perception suggest that we view and process visuals way more efficiently than written text or written data. In fact, according to the Social Science Research Network, 65% of human beings are visual learners; and that is why visualizing data of any form makes so much more sense.

**5.3 Summary**

This chapter provided all the necessary details of the methodology of our system.

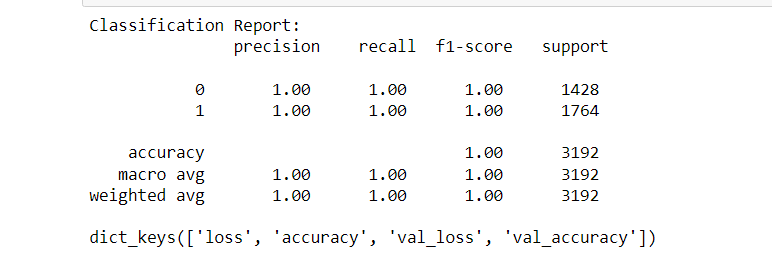
**CHAPTER 6**

**DESIGN RESULTS AND ANALYSIS**

**6.1 Introduction**

After training Deep feature extraction has been carried out using deep CNN models that have already been trained, including ResNet50, InceptionV3, MobileNetV2, and VGG16. The accuracy of the classifications was used as the performance indicator in this study. The pre-trained tailored models, such as MobileNetV2, InceptionV3, and VGG16, each achieved 98% accuracy while the developed model achieved 93% accuracy and 98% validation accuracy. However, our model provides findings that are nearly 100% accurate, which is a remarkable accomplishment given that no other solution to this problem has produced results with this level of accuracy.

**6.2 Result Achieved**



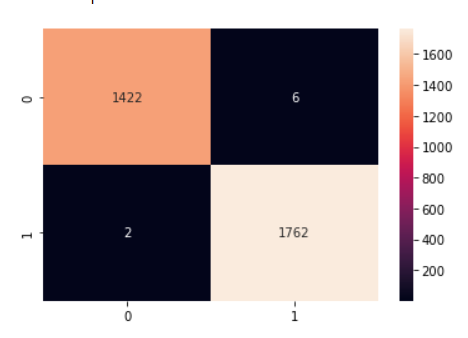
**Figure 6.1:** Classification Report of the entire model

Here,the high accuracy we claim about our model is shown below in the classification report, where we can see that our model is giving exact 100% accuracy [since, f1-score is equal to 1.00].



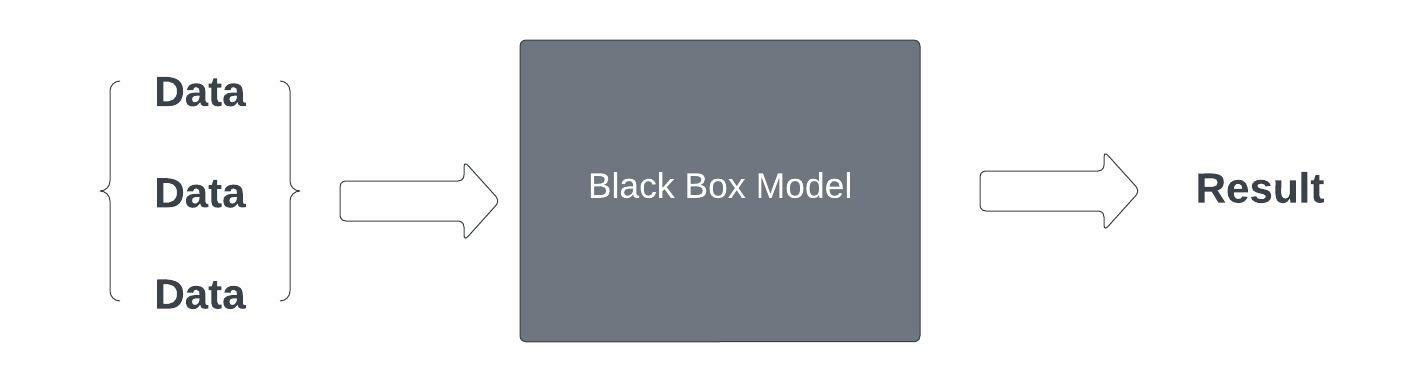
**Figure** **6.2:** Accuracy and Loss graph

Our model provided 99.19% accuracy and 0.81% loss in the 50th epoch after training using the train generator, validation generator, and 50 epochs. The training accuracy was fairly low in the first few epochs, starting at 57.42% and increasing to 99.19% after the 50th epoch



**Figure 6.3:** Confusion Matrix

From this confusion matrix, we can see our model can predict output with almost 100% accuracy, since the value of a true positive here is 1422, a false positive is 6, a false negative is 2, and a false positive is 1762.

  
**Figure 6.4**: Deep Learning model as a Black-Box

When the data point is fed, **LIME** generates a new dataset consisting of permuted samples and the corresponding predictions of the black-box model. An interpretable model is trained on this new dataset, which is weighted by the proximity of the sampled instances to the instance of interest. This is how we get to know about the reason behind the model making predictions by making variations to the data being fed into the black-box as input.

**Model Evaluation**

The models' performance is assessed using accuracy, precision, recall, and the F1-score. The suggested model's performance was evaluated using the terms true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The rate at which damaged photos are directly detected from all images is referred to as "recall," also known as "sensitivity." Precision's recall is comprehensive. The F1-score is a combination of accuracy and recall that indicates how often the anticipated value is correct. In mathematics, it is also known as the harmonic mean of p and r. The equations are shown below. Matrixes may be used to evaluate the performance of a system and, after the construction of the model, its performance. Accuracy is a measure of how well a model predicts the given input. (i.e., how many times the model accurately predicts the actual outcome). The mathematical formulas used to determine accuracy are provided in equations 1 and 2

accuracy=(TP+TN)/(TP+TN+FP+FN) (1)

accuracy=correctpredictions/totalnumberofexample. (2)

The rate of correctly recognizing the true value from a set of all values is known as recall, also known as sensitivity. The expression can be used to calculate recall in Eq. (3).

re-call=TP/(TP+FN) . (3)

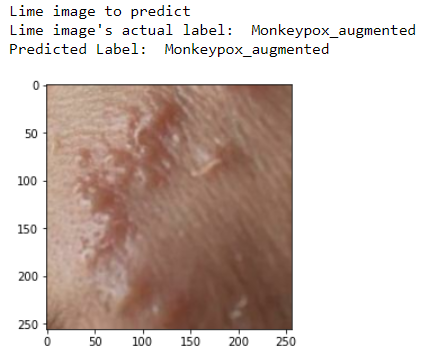
Precision is defined as the number of correct identifications. The number of times the model's positive forecast was correct may be computed using the following mathematical formula, which is more connected to the model's identification in Eq.

precision=TP/(TP+FP). (4)

A single matrix may be used to describe the classifier's performance for both recall and precision, and the F1-score is a single matrix that measures precision and recall. In other words, it is also known as a harmonic method of precision and recall. The F1-score is calculated using Equation (5).

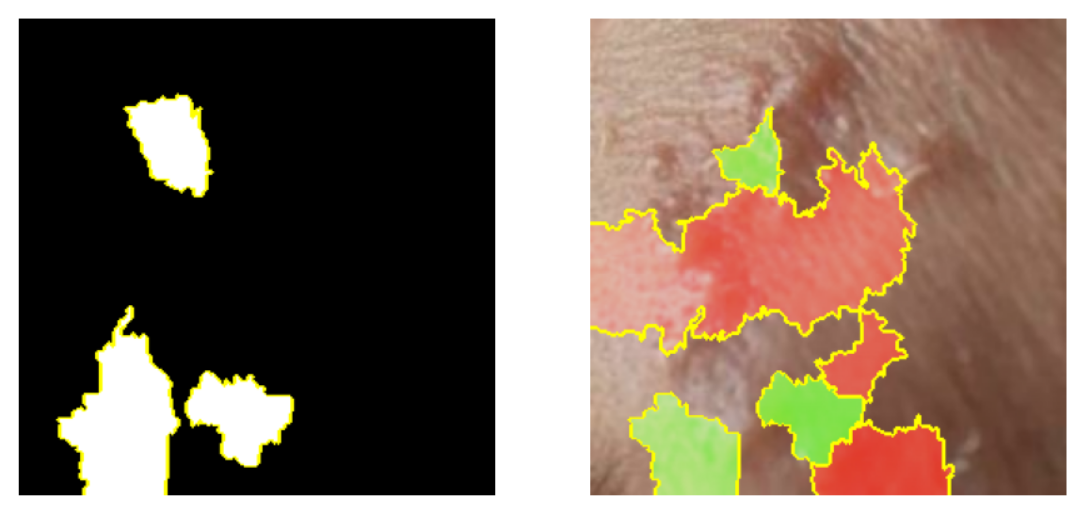
f1-score=2pr/(p+r). (5)

**Figures and Tables**

****

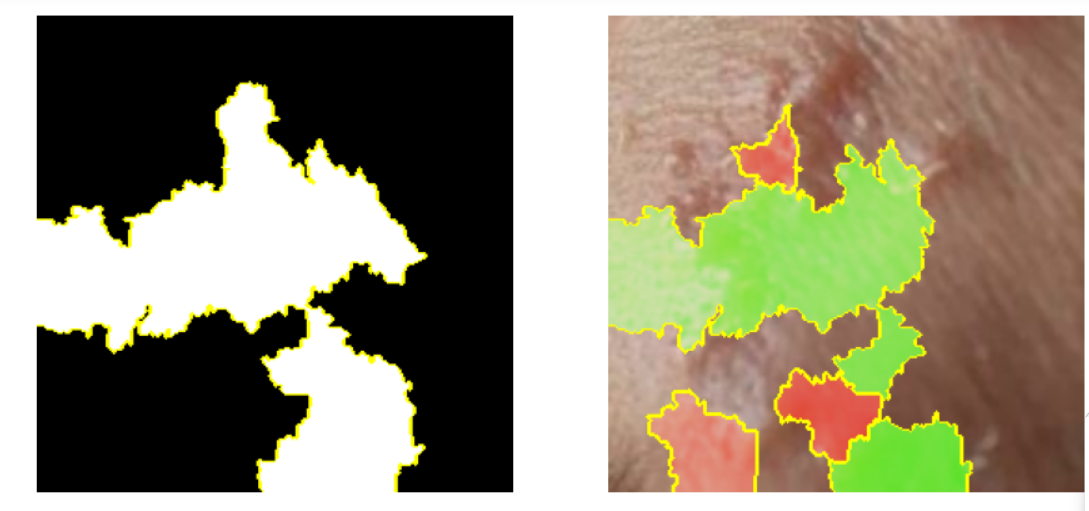
**Figure 6.5(a):** Prediction of Monkeypox.

A test prediction was made to see if the image was of a monkeypox. Lime is used as a tool for explainable AI prediction in order to interpret the regions indicating monkeypox.

****

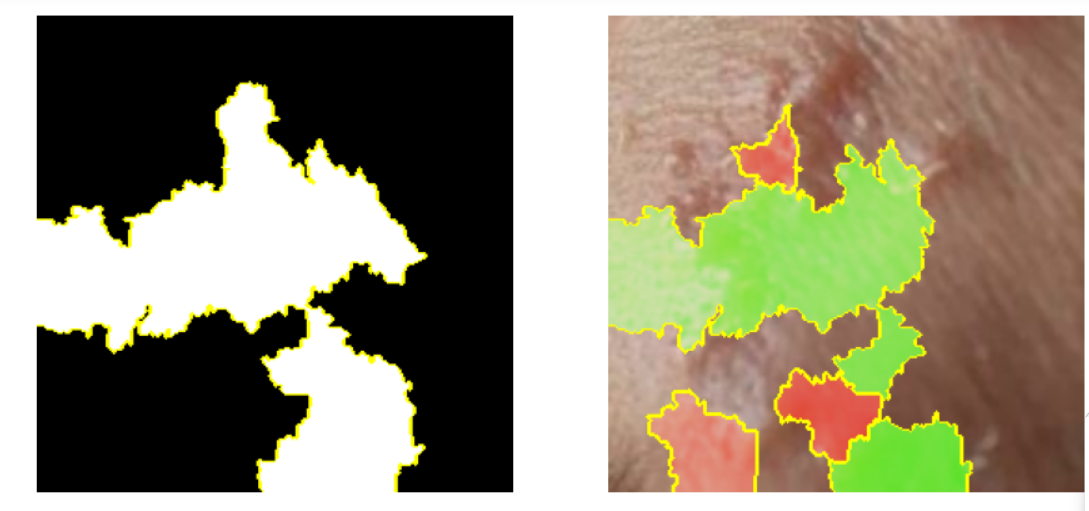
**Figure 6.5(b):** Lime Image Interpretation with Bounded Regions.

We can find the boundaries that represent the regions of predicting each class using lime. The green boundaries represent the region most responsible for the predicted label, while the red boundaries represent the regions that are not responsible for that prediction, even if they appear the same.

****

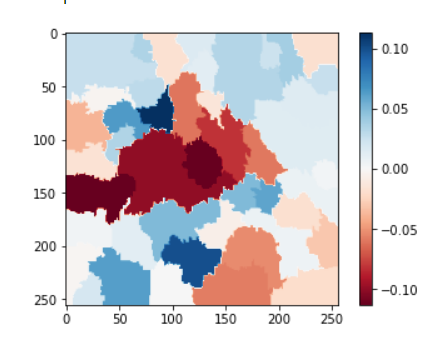
**Figure 6.5(c):** Lime Image Interpretation with Boundaries Overlapping each other strength wise.

The beauty of lime within the favored regions shows which portion is more likely to give that output by simply increasing the density of the green colored region while interpreting the favored and not favored regions.

****

**Figure 6.5(d):** Lime Overall Bounded Regions for Interpretation.

Lime displays an overall region of affected area where the positive region is, in our case, the monkeypox region. It simply shows the overall bounded region without the unnecessary regions.

****

**Figure 6.5(e):** Lime Heat map of the Predicted Image

Using Lime, we are able to generate a heat map of the previous predicted image output. Where the blue regions are the regions that are most likely to represent monkeypox, on the other hand, the red regions are the regions that are not monkeypox’s region. A scale of the heat map is shown, representing the strength of those red and blue regions. A positive-scaled region has positive output and, in our case, monkeypox, whereas a negative scale does not have monkeypox. where 0.00 indicates that the region is not responsible for monkeypox or any other pox, but is simply an irrelevant region. Whereas 0.10 or higher means most susceptible to monkeypox, and -0.10 or lower means not susceptible to monkeypox.

**6.3 Summary**

Our model produces nearly 100% correct results, which is a remarkable feat considering that no previous method to this problem has produced results with this much precision.

**CHAPTER 7**

**SKILLS**

**7.1 Introduction**

In this chapter we discuss the skills that we have obtained in order to develop this massive sophisticated system.

**7.2 Skills obtained**

Through this project the following skills have been developed:

* **Skill in Programming & Tools**
* **Python**

A function is a group of connected claims that together carry out a mathematical, analytic, or evaluative activity. The specific job is returned via a group of statements referred to as Python Functions. Python functions are easy to define and crucial for programming at the intermediate level. The same standards apply to variable names and function names. The objective is to define a function by grouping together a few often performed actions. Instead of continually constructing the same code block for different input variables, we may call the function and reuse the code it contains. The two primary types of functions in Python are user-defined and built-in. It keeps the program focused, distinctive, and well-organized.

* **Matplotlib**

A Python package called Matplotlib is a numerical and mathematical addition to the NumPy library. A Matplotlib module that offers a MATLAB-like interface is accessible through Pyplot, a state-based interface. Line plots, contour plots, histogram plots, scatter plots, 3D graphs, and more may all be utilized with Pyplot.

* **TensorFlow**

You may adopt best practices for data automation, model tracking, performance monitoring, and model retraining with the help of the TensorFlow platform. Success depends on the use of production-level technologies to automate and monitor model training over the course of a good, service, or business process.

* **Scikit-learn**

In the Python ecosystem, Scikit-learn, an open source data analysis toolkit, is considered to be the pinnacle of machine learning (ML). Important ideas and traits include: algorithms for making decisions, such as: Data are identified and categorized by classification based on patterns.

* **Deep Learning**

Deep learning is a machine learning method that instructs computers to learn by doing what comes naturally to people. Driverless cars use deep learning as a vital technology to recognize stop signs and tell a pedestrian from a lamppost apart.

**7.3 Summary**

In this chapter we discussed the list of skills that have been obtained throughout the process of developing and materializing this system.

**CHAPTER 8**

**FUTURE WORK**

**8.1 Introduction**

This chapter discusses the future scope or the implementation of this system. As our system is a web-based, various forms of new features can be incorporated to this system as per the requirements.

**8.2 Future Scope of Work**

The main objective of developing this system is to provide a basic platform for startup clinics and diagnostic centers. The system can be more improved than the current form. In future we will update our dataset by collecting data by ourselves, then the system will work more precisely. And our plan is to sell our this project to the hospitals and clinics, by using our system they will detect recognize that if the is affected from monkeypox or not and that will be very time saving as well as cheap.

**8.3 Summary**

This chapter has described the possible future applications of the design. But there are a lot of possibilities with the designed system. The system may need some research for different applications, though the principle of the designed system will remain as it is.

**CHAPTER 9**

**DESIGN IMPACT**

**9.1 Introduction**

In this chapter, we discuss about the various impacts that our system has been able to generate.

**9.2 Environmental Impact**

By introducing this system in a diagnostic center huge amounts of paper can be saved because all the tasks will be computerized therefore it would the number of paper wasted per day in official works can be hugely eradicated.

**9.3 Economic Impact**

The economic impact that this system entails is that by introducing this system in an organization, employees’ salary can be reduced significantly as there would now require less man power to complete a certain task because the system is now automated and requires less human effort.

**9.4 Social Impact**

Our model can have a positive impact on our society as well as bring revolutionary changes to medical science by generating output at this satisfactory level. Our model can create the smallest possible mask that changes the model’s original prediction on an image when given a perturbed image (that same image with the mask applied). And, by implementing heatmap data visualization, we can view and process visuals way more efficiently than written text or written data.

**9.5 Sustainability**

Our system has been able to deal with huge number of patients’ information at a time. When the numbers of medical tests are conducted simultaneously our system remains stable. Therefore based upon these facts and continuous testing, our system is sustainable.

**9.6 Summary**

This chapter has covered the different types of impacts that our system offers and those has been described and discussed. From the above given impacts we can conclude that our designed system is good enough to use under any circumstance.

**CHAPTER 10**

**COMPLIANCE WITH IEEE STANDARDS**

**10.1 Introduction**

In this section we discuss about the consistence of our task with diverse standards. There are a few distinct standards, amongst which the IEEE standards, US standards and European standards are talked about in this part.

**10.2 Compliance with IEEE standard**

There are a few distinct guidelines put forward by IEEE Standards affiliation. The majority of them however are not material for our framework. We have included idea of operation as for the IEEE standard. A conference paper has additionally been submitted and affirmed by IEEE standards entitled "Monkeypox Detection using Deep Learning and Explainable AI (XAI)" that points of interest out our work on this task.

**10.3. Compliance with US standard**

ANSI recommends that copyrighted software should only be included for informational purposes, or in forms which do not mandate particular implementations of the standard. Object code should never be included in a standard as a normative requirement. While ANSI opposes use of software standards to mandate particular implementations and believes that use of software in standards should be avoided to the extent possible, ANSI recognizes that there may be circumstances in which inclusion of some software, provided it is accompanied by adequate legal permissions, may facilitate development of multiple, competing and interoperable implementations of the standard. Examples of such software could include: ·

* Pseudo Code (code that is human readable and similar to programming languages but cannot be directly processed or compiled directly to be processed by hardware that manipulates data according to instructions);
* Schema examples;
* Data structure definitions;
* ASN.1 structure definitions;
* ABNF grammar specifications;
* Example programming instructions that are sufficiently limited in scope that they do not, either singularly or in the aggregate, perform a complete or a substantial part of a function and are illustrative, at most, of limited sections of an independent fully described specification; or
* Sample programming instructions provided solely for conformance testing purposes.

Our project has been established based on the above ANSI principles and it completely relies upon it.

**10.4 Summary**

In this section we have examined the different compliant standards and made sure that we are in accordance with. These standards have been put without hesitation so as to control things, guarantee well-being and ensure there are no well-being dangers to the use of distinctive segments. It is imperatively essential to maintain these measures and we have done as such over the span of our task work.

**CHAPTER 11**

**CONCLUSION**

In this project we achieved our goal by developing a model that can accurately predict whether a patient's skin disease is caused by infectious monkeypox or not. The model we develop will not only assist in determining whether it is monkey pox or not, but it will also show possible regions that may cause monkey pox, increasing the reliability of our model in the medical field. Another important goal of our model we accomplished here was to make it user-friendly and marketable in the context of Bangladesh medical services. To train our multiple models for accurate prediction, we use a variety of approaches, such as creating some CNN layers after setting the parameters and batch size, and for better accuracy, setting the epoch to 50, using an AI tool like Lime, explainable masking method for image classifiers, and heatmap data visualization. After training, our model produces nearly 100% correct results, which is a remarkable feat considering that no previous method to this problem has produced results with this much precision. Overall, by producing output at this reasonable level, our approach may have a good influence on our society as well as bring revolutionary advances to medical research. When given a disturbed picture, our model can generate the lowest feasible mask that modifies the model's original prediction on the image (that same image with the mask applied). Furthermore, by utilizing heat map data visualization, we can examine and analyze images far more effectively than written text or written data.Future work:

**BIBLIOGRAPHY**

1. Moore, M.J.; Rathish, B.; Zahra, F. Monkeypox. In StatPearls; StatPearls Publishing: Treasure Island, FL, USA, 2022.
2. Nolen, L.D.; Osadebe, L.; Katomba, J.; Likofata, J.; Mukadi, D.; Monroe, B.; Doty, J.; Hughes, C.M.; Kabamba, J.; Malekani, J.; et al. Extended Human-to-Human Transmission during a Monkeypox Outbreak in the Democratic Republic of the Congo. Emerg. Infect. Dis. **2016**, 22, 1014–1021.
3. Nguyen, P.Y.; Ajisegiri, W.S.; Costantino, V.; Chughtai, A.A.; MacIntyre, C.R. Reemergence of Human Monkeypox and Declining Population Immunity in the Context of Urbanization, Nigeria, 2017–2020. Emerg. Infect. Dis. 2021, 27.
4. World health organization (2022) Multi-country monkeypox outbreak: situation update. https:// www. who. int/ emergencies/ disea se- outbreak- news/ item/ 2022- DON396. (Accessed: 2022-06-30)
5. Nguyen, P.Y.; Ajisegiri, W.S.; Costantino, V.; Chughtai, A.A.; MacIntyre, C.R. Reemergence of Human Monkeypox and Declining Population Immunity in the Context of Urbanization, Nigeria, 2017–2020. Emerg. Infect. Dis. **2021**, 27
6. Hayward, G. Remarks on Smallpox, Cowpox and Varioloid. Boston Med. Surg. J. 1860, 62, 173–177.
7. Islam, T.; Hussain, M.A.; Chowdhury, F.U.H.; Islam, B.M.R. AWeb-scraped Skin Image Database of Monkeypox, Chickenpox, Smallpox, Cowpox, and Measles. bioRxiv 2022.
8. ChiranjibiSitaula ;TejBahadurShahi, Monkeypox Virus Detection Using Pre‑trained Deep Learning‑based Approaches
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
10. Sarumi OA (2020) Machine learning-based big data analytics framework for ebola outbreak surveillance. In: International Conference on Intelligent Systems Design and Applications, Springer, pp 580–589
11. Sandeep R, Vishal K, Shamanth M, Chethan K (2022) Diagnosis of visible diseases using cnns. In: Proceedings of International Conference on Communication and Artificial Intelligence, Springer, pp 459–468
12. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations
13. Lara JVM, Velásquez RMA (2022) Low-cost image analysis with convolutional neural network for herpes zoster. Biomedical Signal Processing and Control 71:103250
14. MdManjurul Ahsan, Muhammad Ramiz Uddin, MithilaFarjana, Ahmed NazmusSakib, Khondhaker Al Momin and ShahanaAkter Luna, Image data collection and implementation of deep learning-based model in detecting monkeypox disease using modified VGG16.
15. Shams Nafisa Ali, Md. Tazuddin Ahmed, Joydip Paul, TasnimJahan, S. M. SakeefSani, Nawsabah Noor, TaufiqHasan, Monkeypox Skin Lesion Detection Using Deep Learning Models: A Feasibility Study] [https://arxiv.org/abs/2207.03342
16. Abdelhamid, A.A.; El-Kenawy, E.-S.M.; Khodadad, N.; Mirjalili, S.; Khafaga, D.S.; Alharbi, A.H.; Ibrahim, A.;Eid, M.M.; Saber, M. Classification of Monkeypox Images Based on Transfer Learning and the Al-Biruni Earth Radius Optimization Algorithm.
17. Jason Brownlee, Deep Learning Performance, Gentle Introduction to the Adam Optimization Algorithm for Deep Learning, Machine Learning Mastery, July 3, 2017, Available: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
18. O. A. Hamid, L. Deng, D. Yu, &#39;&#39;Exploring convolutional neural network structures and optimization techniquesfor speech recognition,”ISCA,Vol. 11, pp. 73-5, 2013. Available: https://www.microsoft.com/en-

us/research/publication/exploring-convolutional-neural-network-structures-and-optimization-techniques-for-speech- recognition/

1. J. Brownlee, &quot;A gentle introduction to pooling layers for convolutional neural networks&quot;, Machine Learning Mastery, 2021. Available: <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>
2. J. Jeong, &quot;The most intuitive and easiest guide for CNN,&quot; Medium, 2021. Available:https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480
3. S. Saha, &quot;A comprehensive guide to convolutional neural networks — the ELI5 way,&quot; Medium, 2021.Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-> way-3bd2b1164a53
4. ”Explainable AI,” The Royal Society, 28 November 2019. (link resides outside IBM) . Available: https://www.ibm.com/watson/explainable-ai
5. ”Explainable Artificial Intelligence,” Jaime Zornoza, 15 April 2020. (link resides outside IBM)
6. ”Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI,” ScienceDirect, June 2020. (link resides outside IBM)

**APPENDIX**

**SOFTWARE LISTING**

**Import Dependencies:**

import tensorflow as tf

from tensorflow.keras import models, layers

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

%matplotlib inline

from sklearn.metrics import confusion\_matrix

import seaborn as sns

import itertools

**About Dataset:**

IMAGE\_SIZE = 256

BATCH\_SIZE = 64

CHANNELS = 3

EPOCHS = 50

dataset = tf.keras.preprocessing.image\_dataset\_from\_directory(

"Augmented Images",

seed=123,

shuffle = True,

image\_size = (IMAGE\_SIZE,IMAGE\_SIZE),

batch\_size = BATCH\_SIZE,

)

class\_names = dataset.class\_names

class\_names

len(dataset)

for image\_batch, labels\_batch in dataset.take(1):

plt.figure(figsize = (15,10))

for image\_batch, labels\_batch in dataset.take(1):

for i in range(12):

ax = plt.subplot(3,4, i+1)

plt.imshow(image\_batch[i].numpy().astype("uint8"))

plt.title(class\_names[labels\_batch[i]])

plt.axis("off")

**Divide Train, Test and Validation data:**

train\_size = 0.8

len(dataset)\*train\_size

train\_ds = dataset.take(40)

len(train\_ds)

test\_ds = dataset.skip(40)

len(test\_ds)

val\_size = 0.1 #validation

len(dataset)\*val\_size

val\_ds = test\_ds.take(5)

len(val\_ds)

test\_ds = test\_ds.skip(5)

len(test\_ds)

**Dataset Partitions:**

def get\_dataset\_partitions\_tf(ds,train\_split = 0.8, val\_split = 0.1, test\_split = 0.1, shuffle = True, shuffle\_size = 10000):

assert (train\_split + test\_split +val\_split ) == 1

ds\_size = len(ds)

if shuffle:

ds = ds.shuffle(shuffle\_size, seed = 12)

train\_size = int(train\_split \* ds\_size)

val\_size = int (val\_split \* ds\_size)

train\_ds = ds.take(train\_size)

val\_ds = ds.skip(train\_size).take(val\_size)

test\_ds = ds.skip(train\_size).skip(val\_size)

return train\_ds, val\_ds, test\_ds

train\_ds, val\_ds, test\_ds = get\_dataset\_partitions\_tf(dataset)

len(train\_ds)

len(val\_ds)

len(test\_ds)

resize\_and\_rescale = tf.keras.Sequential([

layers.experimental.preprocessing.Resizing(IMAGE\_SIZE, IMAGE\_SIZE),

layers.experimental.preprocessing.Rescaling(1.0/255),

])

**CNN:**

input\_shape = (BATCH\_SIZE, IMAGE\_SIZE, IMAGE\_SIZE, CHANNELS)

n\_classes = 2

model = models.Sequential([

resize\_and\_rescale,

layers.Conv2D(32, kernel\_size = (3,3), activation='relu', input\_shape=input\_shape),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, kernel\_size = (3,3), activation='relu'),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, kernel\_size = (3,3), activation='relu'),

layers.MaxPooling2D((2, 2)),

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

layers.MaxPooling2D((2, 2)),

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

layers.MaxPooling2D((2, 2)),

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

layers.MaxPooling2D((2, 2)),

layers.Flatten(),

layers.Dense(64, activation='relu'),

layers.Dense(n\_classes, activation='softmax'),

])

model.build(input\_shape = input\_shape)

scores = model.evaluate(test\_ds)

acc = history.history['accuracy']

val\_acc = history.history['val\_accuracy']

loss = history.history['loss']

val\_loss = history.history['val\_loss']

**Using adam optimizer:**

model.compile(

optimizer = 'adam',

loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=False),

metrics=['accuracy']

)

**Accuracy Chart:**

plt.figure(figsize=(8, 8))

plt.subplot(1, 2, 1)

plt.plot(range(EPOCHS), acc, label = 'Training Accuracy')

plt.plot(range(EPOCHS), val\_acc, label = 'Validation Accuracy')

plt.legend(loc = 'lower right')

plt.title('Training and Validation Accuracy')

#loss chart

plt.subplot(1, 2, 2)

plt.plot(range(EPOCHS), loss, label = 'Training Loss')

plt.plot(range(EPOCHS), val\_loss, label = 'Validation Loss')

plt.legend(loc = 'upper right')

plt.title('Training and Validation Loss')

# epoch loss

epochs = range(1, len(acc) + 1)

#plt.plot(epochs, acc,'bo', label='Training acc')

#plt.plot(epochs, val\_acc, 'b', label='Validation acc')

#plt.title('Training and validation accuracy')

#plt.legend()

plt.figure()

plt.plot(epochs, loss,'bo', label='Training loss')

plt.plot(epochs, val\_loss, 'b', label='Validation loss')

plt.title('Training and validation loss')

plt.legend()

plt.show()

import numpy as np

for images\_batch, labels\_batch in test\_ds.take(1):

first\_image = images\_batch[0].numpy().astype('uint8')

first\_label = labels\_batch[0].numpy()

print("First image to predict")

plt.imshow(first\_image)

print("First image's actual label: ", class\_names[first\_label])

#lets do some prediction since model is complete

batch\_prediction = model.predict(images\_batch)

print("Predicted Label: ",class\_names[np.argmax(batch\_prediction[0])])

**Predict Model:**

def predict(model, img):

img\_array = tf.keras.preprocessing.image.img\_to\_array(images[i].numpy())

img\_array = tf.expand\_dims(img\_array, 0) # create a batch

predictions = model.predict(img\_array)

predicted\_class = class\_names[np.argmax(predictions[0])]

confidence = round(100 \* (np.max(predictions[0])), 2)

return predicted\_class, confidence

plt.figure(figsize = (15, 15))

for images, labels in test\_ds.take(1):

for i in range(9):

ax = plt.subplot(3,3, i+1)

plt.imshow(images[i].numpy().astype("uint8"))

predicted\_class, confidence = predict(model, images[i].numpy())

actual\_class = class\_names[labels[i]]

plt.title(f"Actual: {actual\_class},\n Predicted: {predicted\_class}.\n Confidence: {confidence}%")

plt.axis("off")

y\_pred = [] # store predicted labels

y\_true = [] # store true labels

# iterate over the dataset

for image\_batch, label\_batch in dataset: # use dataset.unbatch() with repeat

# append true labels

y\_true.append(label\_batch)

# compute predictions

preds = model.predict(image\_batch)

# append predicted labels

y\_pred.append(np.argmax(preds, axis = - 1))

# convert the true and predicted labels into tensors

correct\_labels = tf.concat([item for item in y\_true], axis = 0)

predicted\_labels = tf.concat([item for item in y\_pred], axis = 0)

**Confusion Matrix:**

cm = confusion\_matrix(correct\_labels, predicted\_labels)

sns.heatmap(cm, annot = True,fmt='g')

**F1 Score:**

from sklearn.metrics import confusion\_matrix , classification\_report

print("Classification Report: \n", classification\_report(correct\_labels, predicted\_labels))

print(history.history.keys())

**Lime:**

from lime import lime\_image

explainer = lime\_image.LimeImageExplainer()

import numpy as np

for images\_batch, labels\_batch in test\_ds.take(1):

lime\_image = images\_batch[0].numpy().astype('uint8')

lime\_label = labels\_batch[0].numpy()

print("Lime image to predict")

plt.imshow(lime\_image)

print("Lime image's actual label: ", class\_names[lime\_label])

#lets do some prediction since model is complete

batch\_prediction = model.predict(images\_batch)

print("Predicted Label: ",class\_names[np.argmax(batch\_prediction[0])])

exp = explainer.explain\_instance(lime\_image,

model.predict,

top\_labels=5,

hide\_color=0,

num\_samples=1000)

**Musking:**

from skimage.segmentation import mark\_boundaries

temp\_1, mask\_1 = exp.get\_image\_and\_mask(exp.top\_labels[0], positive\_only=True, num\_features=5, hide\_rest=True)

temp\_2, mask\_2 = exp.get\_image\_and\_mask(exp.top\_labels[0], positive\_only=False, num\_features=10, hide\_rest=False)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,15))

ax1.imshow(mark\_boundaries(temp\_1, mask\_1))

ax2.imshow(mark\_boundaries(temp\_2, mask\_2))

ax1.axis('off')

ax2.axis('off')

explanation= explainer.explain\_instance(lime\_image,model.predict, top\_labels=2, hide\_color=0, num\_samples=1000)

temp\_1, mask\_1 = exp.get\_image\_and\_mask(exp.top\_labels[1], positive\_only=True, num\_features=10, hide\_rest=True)

temp\_2, mask\_2 = exp.get\_image\_and\_mask(exp.top\_labels[1], positive\_only=False, num\_features=10, hide\_rest=False)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,15))

ax1.imshow(mark\_boundaries(temp\_1, mask\_1))

ax2.imshow(mark\_boundaries(temp\_2, mask\_2))

ax1.axis('off')

ax2.axis('off')

from skimage.segmentation import mark\_boundaries

temp, mask = explanation.get\_image\_and\_mask(explanation.top\_labels[0], positive\_only=True, num\_features=5, hide\_rest=True)

plt.imshow(mark\_boundaries(temp / 2 + 0.5, mask))

temp, mask = explanation.get\_image\_and\_mask(explanation.top\_labels[0], positive\_only=True, num\_features=5, hide\_rest=False)

plt.imshow(mark\_boundaries(temp / 2 + 0.5, mask))

temp, mask = explanation.get\_image\_and\_mask(explanation.top\_labels[0], positive\_only=False, num\_features=10, hide\_rest=False)

plt.imshow(mark\_boundaries(temp / 2 + 0.5, mask)

#Select the same class explained on the figures above.

ind = explanation.top\_labels[0]

#Map each explanation weight to the corresponding superpixel

dict\_heatmap = dict(explanation.local\_exp[ind])

heatmap = np.vectorize(dict\_heatmap.get)(explanation.segments)

#Plot. The visualization makes more sense if a symmetrical colorbar is used.

plt.imshow(heatmap, cmap = 'RdBu', vmin = -heatmap.max(), vmax = heatmap.max())

plt.colorbar()