GENETIC BASED DISEASE IDENTIFICATION WITH DEEP LEARNING ON NEURAL NETWORKS

Madhulika Dayal
700743206
dept.Computer Science
University of Central Missouri
mxd32060@ucmo.edu

Akhilandeswari Vegi
700758173
dept.Computer Science
University of Central Missouri
axv81730@ucmo.edu

Nikhil reddy Kethireddy
700739505
dept.Computer Science
University of Central Missouri
nxk95050@ucmo.edu

Srusti Katla
700717867
dept.Computer Science
University of Central Missouri
sxk78670@ucmo.edu

***Abstract*— The identification of genetic disorders remains a challenging task in medical research, despite the use of deep learning techniques. The current methods for identification have accuracy issues, which are further complicated by the lack of labelled and unlabelled data. To tackle this issue, semi-supervised approaches like label propagation and positive-unlabelled learning are leveraged to identify candidate disease genes, using unknown genes for training. Recent advancements in deep learning and diagnostic imaging have revolutionized computerized healthcare. Deep learning techniques have opened up new possibilities in multimedia healthcare distribution. To enable early detection of genetic diseases, a study suggests utilizing an Improved Algorithm and statistically significant text information. Key clinical text information such as age, sex, genes is incorporated, while resting-state functional data is used to measure brain connectivity. Deep learning techniques are used for data interpretation and analysis, enabling the classification of variations and data models. This approach will offer the better solutions to identifying genetic disorders in real-world data. The Convolutional Neural Network algorithm is particularly will be effective in checking data in a more compact way with training and testing data, delivering results.**

This promising development in the field of healthcare can lead to early diagnosis and prevention of genetic diseases. The application will be developed with Google Colab Python Tool as the project can be directly executed in any type computer systems with internet connection. There is no need of any specific software to be installed in the user system. The Colab Tool helps to develop and run the application directly inside the cloud server where the Python library files are installed. The deep learning algorithm libraries are built inside the Colab. It helps the project to use the deep learning algorithm in the finding of Genetic disease.

https://github.com/Madhulika014/Final-Project

***Keywords*— Genetic disease, Convolutional Neural Networks, Auto Encoders, health care, genetic**

I. INTRODUCTION

The identification of gene-disease associations is of immense importance in the diagnosis and treatment of human diseases. However, the number of disease-related genes that have been identified and reported in public databases such as the Web-based and the Genetic Association Database is limited. Therefore, the search for disease genes is still crucial. Traditional gene mapping techniques involve linkage analysis and comprehensive association studies. However, due to the limited number of crossovers in tested families, linkage analysis only identifies chromosomal regions that may contain dozens or even hundreds of candidate genes. Comprehensive association studies may also identify multiple regions that still need to be investigated in future research, making experimental validations of many candidate genes time-consuming and expensive.

Since integrating multiple sources of data is essential for identifying disease-related genes, a series of network-based computational approaches have been proposed over the past decade. The common idea behind these methods is that genes causing similar or related diseases will be closely related to each other in biological networks. These models typically use text-mining of biomedical literature, functional annotations, pathways and ontologies, co-expression relationships, intrinsic gene properties, protein interactions, regulatory information, orthologous relationships, and gene expression data to identify candidate disease genes[1].

For example, the text-mining approach to identify a large number of human gene networks contained in the database. The gene-disease associations by using a global network distance measure called random walk analysis will define similarities in protein-protein interaction networks. However, the main limitation of these network-based methods is that they fall short of summarizing complex diseases, for which there are no gene linkage studies yet. The potential of Deep Learning lies in the belief that we can replicate the functioning of the human brain by creating the right connections using silicon and wires, much like living neurons and dendrites [2].

The human brain is a complex network of 80 billion nerve cells known as neurons that are connected to thousands of other cells through Axons. These neurons receive inputs from the surrounding environment or physical organs through dendrites, which generate electrical signals that quickly travel through the brain network. Based on these inputs, a neuron can either transmit the message to another neuron to address the issue or choose not to send it.

To address this issue, another approach called Inductive Grid Completion would be applied, which is based on multiple biological sources and can be implemented to diseases not observed at training time. Of all the methods for identifying genes relevant to a given disease, the traditional Integrated Network-based Deep Learning approach will perform the best in identification of genetic disease.

A genetic disorder is a medical condition commonly caused by mutations in DNA or changes in the overall structure or number of chromosomes. Hereditary gene mutations are known to cause several types of well-known diseases. Genetic testing plays a crucial role in aiding patients to make informed decisions regarding the prevention, treatment, or early detection of hereditary disorders. With the growing population, studies show an exponential increase in the incidence of genetic disorders. While genetic disorders primarily affect physical health, they also impact the psychological and social well-being of patients and their families. Since genetic disorders are chronic conditions that require constant attention but lack cures or treatments, they have powerful effects on families [3].

Genetic diagnoses for one family member may have implications for the health of relatives, even without any current symptoms. Precision genomics-based medicine has emerged as a means of providing personalised and effective treatment based on patients' genetic characteristics. Researchers are aiming to capitalise on advances in genomics to develop increasingly accurate illness risk prediction models to realise the full potential of precision medicine.

The model, derived from a predictor and two classifiers, predicts the presence of genetic disorders and specifies the disorder and disorder subclass, if present. Despite recent progress in polygenic risk scores, the outcomes of these scores are still limited due to present methodologies.

II. MOTIVATION

The primary motivation of Genetic Disease Identification analysis with Deep Learning on Neural Networks is to detect the find out the genetic disease in the hospital dataset. In this work, the dataset containing the patient dataset will be taken into consideration. The pre-processing will be applied in to the dataset and the noisy and null value data will be removed from the dataset. After the data will be analysed and visualized for further processing. The Convolutional Neural Networks algorithm will be chosen to implementation process. The project evaluation can be tested with the deep learning algorithm prediction results. Since the Convolutional Neural Networks algorithm will be used to predict the genetic disease, the accuracy of the algorithm result will be helpful to evaluate the results. The accuracy score of the algorithm in the Genetic Disease Identification helps to evaluate the dataset.

The Deep learning will be the python based application which contributes to find out the Genetic disease early stage. It will be helpful for the human to detect at early and to take necessary treatments in the correct time. The progression of profound learning influences is generally applied to classification assignments and portrayals learning. These profound frameworks with numerous layers have been displayed to yield promising execution in removing serious areas of strength for more of information. The streamlining of the goal capability becomes curved in the event that we adjust one variable and fix the others.

The finding of the application includes the 'Clinical Elements' and 'Clinical Administration' segments of the website pages that report the side effects, prescription and reactions by patients, and related investigations of impacts of various courses of treatments.

Since cross-approval on review information presumably prompts overoptimistic results, cross-approval is improper for this issue. To assess the capacity of the models to anticipate newfound affiliations, we train and test the dataset.

Thus, via consistently consolidating the model for helper side data and the cooperative filter for the quality sickness affiliations grid, our model learns a significantly more significant portrayal for every quality and illness and gives more exact expectation. The project evaluation can be tested with the deep learning algorithm prediction results. Since the Deep learning algorithm will be used to predict the disease, the accuracy of the algorithm result will be helpful to evaluate the results. [4]The accuracy score of the algorithm in the Genetic disease identification helps to evaluate the dataset.

III. CONTRIBUTION & OBJECTIVE

- The objective of Genetic disease identification with deep learning is to detect the Genetic disease in the early stage itself with the available attributes.
- In this work, the dataset containing the hospital patient dataset will be taken into consideration.
- The primary contribution is to apply the deep learning to detect the genetic disease.
- The pre-processing will be applied in to the dataset and the noisy and null value data will be removed from the dataset.
- After the data will be analysed and visualized for further processing.
- The Deep Learning neural network algorithm will be chosen to make the good accuracy prediction.
- It will be helpful in all the hospital patient records to detect the genetic disease.
- The aspect of correlation coefficient data is less sensitive to disease compared to the genetic dataset.
- Distinct attributes of genetic disease occurrence can be removed on varying scales to achieve greater accuracy in performance.

IV. RELATEDWORK

A highlight determination strategy was implemented to decrease the number of atomic descriptors in a fair and unbiased manner. The strategy involved two stages, namely statistical analysis and Genetic Algorithm. In the first stage, descriptors with low standard deviation or containing similar values over half were removed. Subsequently, a Pearson correlation analysis was carried out to determine the relationship among the descriptors and between the descriptor and the target. This step was performed to reduce bias and eliminate descriptors with redundant information. Descriptors with weak correlation with the target (correlation < 0.1) or strong correlation with another objective (correlation > 0.9) were removed [5].

In the case of overlapping descriptors, the one with a weaker correlation with the target was eliminated. In the second stage, a combination of descriptors was chosen using the Genetic Algorithm technique. This technique follows Darwin's principles of natural evolution and uses random methods to obtain optimal non-random solutions. The descriptor selection by GA was performed by defining the solution as a collection of a whole number value in a chromosome. In this case, the number of the value is equal to the number of the selected descriptor, where the value represents the descriptor list. The cross-entropy loss was used as a performance metric during the feature selection [6].

The prediction model uses the Genetic Algorithm technique, which resembles the structure and function of the natural neural system. The primary principle of the Genetic Algorithm is the implementation of artificial neurons, which are simple mathematical models. Such a model has three simple sets of rules, namely reproduction, mutation, and activation. The y-scrambling analysis ensures that the performance of the model did not correlate with an accidental relationship. This analysis was conducted by shuffling the class centre while preserving the descriptors multiple times. The results of the y-scrambling demonstrate by providing the values for shuffled and unshuffled data [7] [8].

V. PROPOSED SYSTEM

The proposed methods aim to find the genetic disease with higher standard. The accuracy levels of the identification of the genetic disease will be improved with the proposed system. The deep learning on neural network will provide the better solution to solve the problem of identification of the genetic disease in the real world hospital data. The Convolutional Neural Network algorithm will check the data in more compact with training and testing the data. It will provide more accuracy as compared with the other type of techniques. The genetic patient dataset will be taken as the input to the application and the dataset will be passed into the Convolutional Neural Network algorithm and the data will be analysed with the different visual graphs.

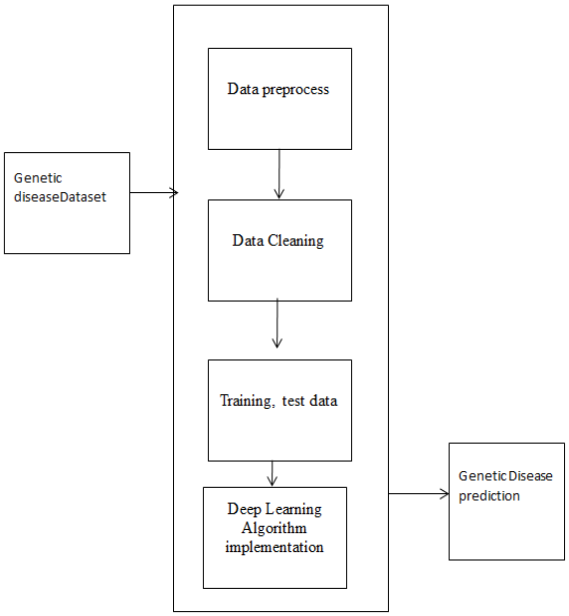

Fig 1 Proposed Architecture diagram

The proposed approach applies a biased neural network function with effectiveness, enabling reliable recognition of genetic disease. The Genetic disease dataset is given as input to the application and the pre-processing is applied , next the data cleaning is performed after the training and test data are split down and will be passed into Deep Learning algorithm and the Genetic disease will be predicted. The Genetic based disease identification with deep learning on neural networks will be the python based application which contributes to find out the genetic disease. It will be helpful in finding of the genetic disease based on the attributes of the patient records.

The testing and training variables are split and passed into the algorithm for the Genetic disease prediction. In this algorithm will provide a comprehensive and intelligent solution for discovering high utility item sets, enabling users to access important information and streamline their search processes.

The application will be developed with Google Colab Python Tool as the project can be directly executed in any type computer systems with internet connection. There is no need of any specific software to be installed in the user system. The Colab Tool helps to develop and run the application directly inside the cloud server where the Python library files are installed. The deep learning algorithm libraries are built inside the Colab.

VI. DATA DESCRIPTION

The dataset for genetic disease identification is taken from the source of kaggle dataset. This dataset contain the fields needed for the analysing of the patient dataset. Exploratory examination is a cycle to investigate and comprehend the information and information relationship in a total profundity with the goal that it makes highlight designing and deep learning demonstrating steps smooth and smoothed out for expectation. Exploratory examination assists with validating our presumptions or misleading. Most of the image in a dataset are noisy and contain lots of information. But with feature engineering do, will get more good results. The first step is to import the libraries and load data. After that will take a basic understanding of data like its shape, sample, is there are any NULL values present in the dataset. Understanding the data is an important step for prediction or any deep learning project. It is good that there are no NULL values.

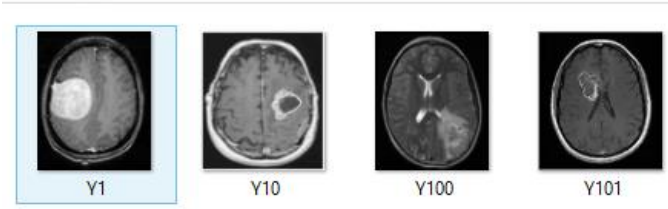

Fig. 2 Sample training dataset

The information about the genetic patient records with different types of attributes are collected from kaggle data. The dataset total contains of image dataset with training and testing genetic affected images of patients. It will begin from the principal segment and investigate every section and comprehend what influence it makes on the objective segment.

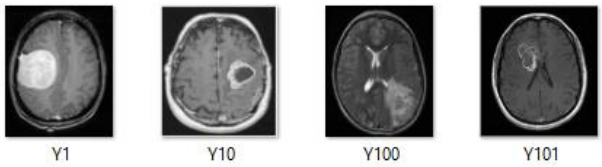

Fig. 3 Sample testing dataset

The dataset of brain x-ray images are downloaded from kaggle website. It has 3 folders train, test, Val which has genetic affected, non-affected patient brain x-ray images.

At the necessary step, we will likewise perform pre-processing and include designing undertakings. The point in acting top to bottom exploratory examination is to get ready and clean information for better Deep Learning demonstrating to accomplish elite execution and summed up models. So it should begin with breaking down and setting up the dataset for expectation.

VII. EXPERIMENTAL ANALYSIS

The Initial process of loading the dataset into the Google Colab into the drive is the first step in execution process. The image data containing the information of the image with respect to the path and the description of the image location and the image related style are linked.

Fig. 4 Load dataset

The pre-processing is applied to the dataset where all the noisy data are removed and the image is reshaped as per the mapping of 255pixel.

The information has an extremely straightforward design with elements. Each folder is related with the Genetic disease brain x-ray images.

```
# what is in the image directory
imageIndex = os.listdir(DATASET_PATH+list_directory[index])
# print(type(imageIndex))
head = 10

# collecting some samples in list
sampleImages = []

# showing indices
for i in range (head):
    sampleImages.append(imageIndex[i])
    print(sampleImages[i])

# choosing some samples to observe
fig=plt.figure()
fig.set_figheight(15)
fig.set_figwidth(15)
axis=[]
row = len(sampleImages)/2
col = row+1

for i in range (len(sampleImages)):
    Image_path=DATASET_PATH+list_directory[index]+"/"+sampleImages[i]
    src = cv2.imread(Image_path)
    image = cv2.cvtColor(src, cv2.COLOR_BGR2RGB)
   # axis.append(fig.add_subplot(row, col, i+1))
    subplot_title=sampleImages[i]
```

Fig. 5 Image re-shape

The image is re-shaped with the following the protocol of making the size of the image to 255 pixel range in any format types. The noisy data present inside the image is also removed and improves the image quality which will be more helpful in the application of the prediction of the genetic disease in the dataset.

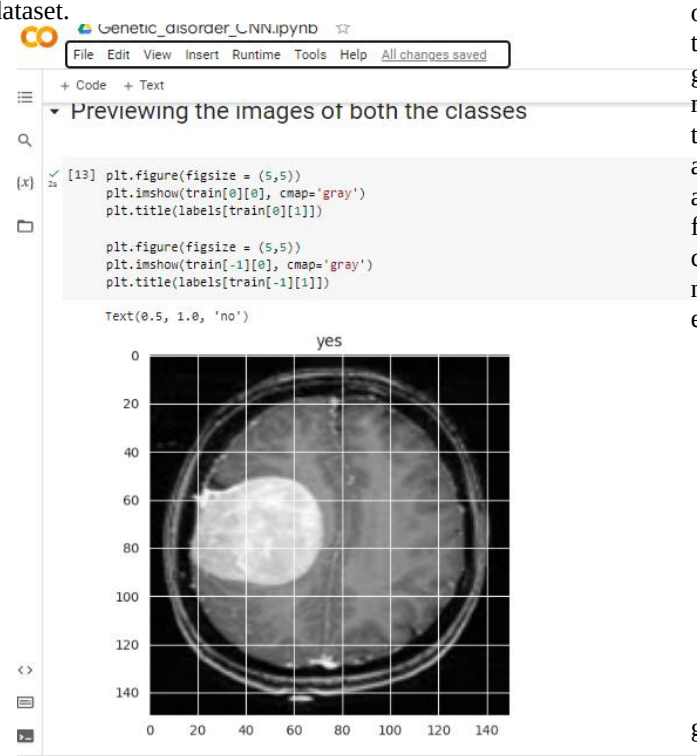

Fig. 6 Genetic image layout

The image dataset is divided into testing and training to pass in to the neural network model.

Training the Model

** CNN **

```
model = Sequential()
model.add(Conv2D(32 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input_sh
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.1))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(128 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(256 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Flatten())
```

Fig. 7 Training and Test data

The Keras Models Programming interface provides a versatile platform for constructing intricate neural networks by adding and removing layers. The API supports both sequential and functional models with a single input and output or multiple inputs and outputs, respectively. The training module encompasses various methods, including generating the model, optimizer, and loss function, fitting the model and evaluating and predicting input data. Furthermore, the API includes methods for batch data processing, testing, and prediction. The Models Programming interface in Keras also enables users to save and pre-process their models for future use. Therefore, this API offers an efficient and comprehensive solution for building and training neural networks. The Keras library files are applied into the execution process.

```
[ ] from keras.models import Sequential
    from keras.layers import Dense

[ ] # https://stats.stackexchange.com/a/136542 helped a lot in avoiding overfitting

    model = Sequential()
    model.add(Dense(11,activation='relu',input_dim=13))
    model.add(Dense(1,activation='sigmoid'))

    model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Fig. 8 Sequential model with keras

The accuracy, confusion matrix of the neural network is given below:

```
[24]
predict_x=model.predict(x_test)
predictions=np.argmax(predict_x,axis=1)

predictions = predictions.reshape(1,-1)[0]
predictions[:15]

1/1 [==============================] - 0s 284ms/step
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Fig. 9 Predictions

The classification report of the prediction is given below:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

[26] print(classification_report(y_test, predictions, target_names = ['Genetic Disorder (

                              precision    recall  f1-score   support

Genetic Disorder (Class 0)       0.50      1.00      0.67         6
           Normal (Class 1)       0.00      0.00      0.00         6

                   accuracy                           0.50        12
                  macro avg       0.25      0.50      0.33        12
               weighted avg       0.25      0.50      0.33        12
```

Fig. 10 Classification Report

Heatmaps use color to display data magnitude, enabling easy identification of patterns and anomalies. Brighter, reddish colors are used to represent more common or higher activity values, while darker colors are used to represent less common values or activity. The shading matrix used to define the heatmap is also commonly referred to as the heatmap itself. To plot heatmaps in Seaborn, simply utilize the seaborn.heatmap() function.

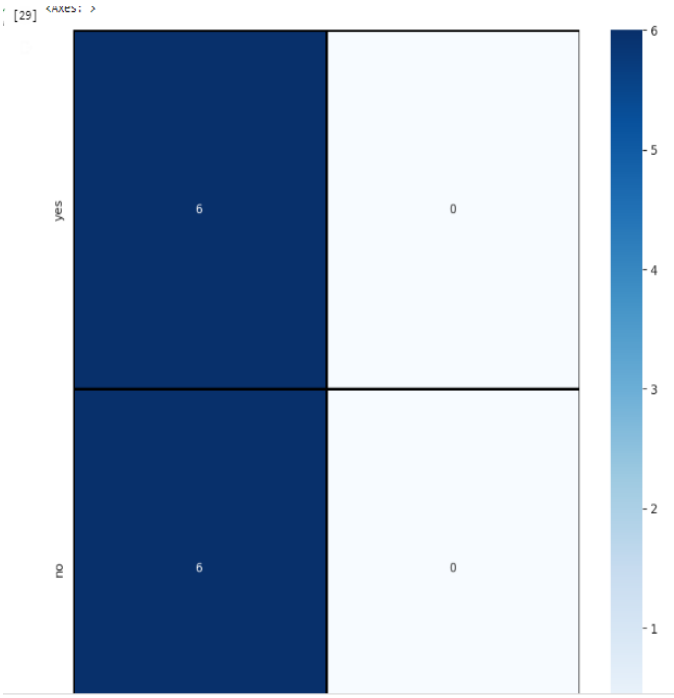

Fig. 11 Heat Map

The heat map matches the genetic diseases is present (yes) and not-present (no) values in the given dataset.

The attributes of input data is been trained and tested then the model is been build, trained and tested and the predicted output is displayed.

Training the Model

** CNN **

```
model = Sequential()
model.add(Conv2D(32 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input_shape
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.1))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(64 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(128 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Conv2D(256 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))
model.add(Flatten())
```

Fig. 12 Sequential data analysis

The results of the CNN with sequential data analysis provides the accuracy. The validation and testing accyracy were identified while executing the convolutional neural network algorithm.

```
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy', patience = 2, ver

history = model.fit(datagen.flow(x_train,y_train, batch_size = 32) ,epochs = 12 , val

Epoch 1/12
1/1 [==============================] - 4s 4s/step - loss: 1.1971 - accuracy: 0.5000 -
Epoch 2/12
1/1 [==============================] - 2s 2s/step - loss: 22.6546 - accuracy: 0.5000
Epoch 3/12
1/1 [==============================] - 1s 1s/step - loss: 11.2785 - accuracy: 0.5000
Epoch 4/12
1/1 [==============================] - 1s 1s/step - loss: 2.4417 - accuracy: 0.7500 -
Epoch 5/12
1/1 [==============================] - 1s 1s/step - loss: 0.7975 - accuracy: 0.7500 -
Epoch 6/12
1/1 [==============================] - ETA: 0s - loss: 0.3305 - accuracy: 0.9167
Epoch 6: ReduceLROnPlateau reducing learning rate to 0.0003000000142492354.
1/1 [==============================] - 1s 1s/step - loss: 0.3305 - accuracy: 0.9167 -
Epoch 7/12
1/1 [==============================] - 1s 1s/step - loss: 0.7858 - accuracy: 0.9167 -
Epoch 8/12
1/1 [==============================] - 1s 1s/step - loss: 0.0602 - accuracy: 1.0000 -
Epoch 9/12
1/1 [==============================] - ETA: 0s - loss: 0.3144 - accuracy: 0.7500
Epoch 9: ReduceLROnPlateau reducing learning rate to 9.000000427477062e-05.
1/1 [==============================] - 1s 1s/step - loss: 0.3144 - accuracy: 0.7500 -
Epoch 10/12
1/1 [==============================] - 2s 2s/step - loss: 0.1689 - accuracy: 0.9167 -
Epoch 11/12
```

Fig.13: Sequential output

The Convolutional Neural Networks algorithm is applied with creating the sequential model. The output of the sequential model with layers is displayed.

```
i = 0
for c in correct[:6]:
    plt.subplot(3,2,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.imshow(x_test[c].reshape(150,150), cmap="gray", interpolation='none')
    plt.title("Predicted Class {},Actual Class {}".format(predictions[c], y_test[c]))
    plt.tight_layout()
    i += 1
```

Fig. 14 prediction class layer

The results of the prediction of genetic disease identification with the CNN provide the accuracy results.

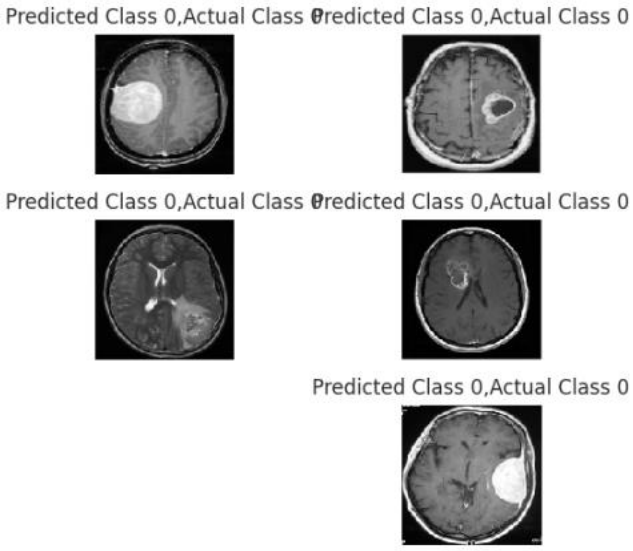

Fig. 15 Prediction Results

The training and validation of the model were evaluated and the accuracy is calculated.

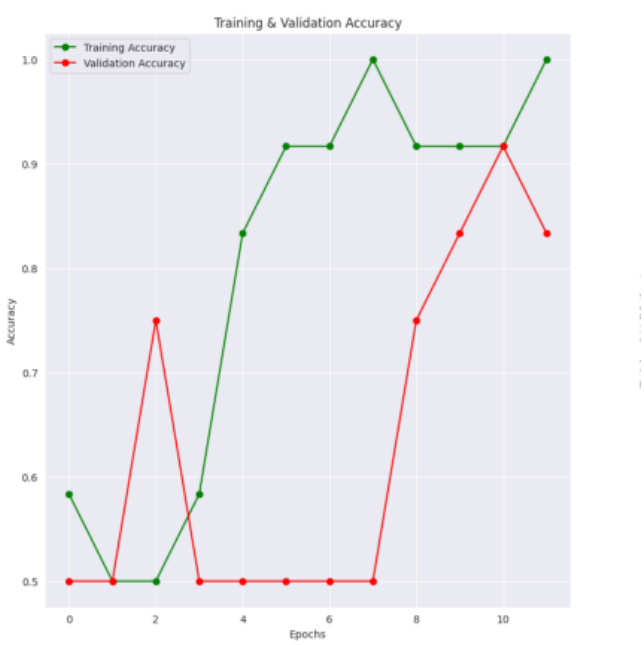

Fig. 16 Training and Validation accuracy

The training and the validation accuracy graph shows the results with graphical format. The trainig accuracy is getting in the increase ratio and it reaches the good saturation point. The validation accuracy is getting in the gradual increase points and reaches the average of 85% of accuracy .

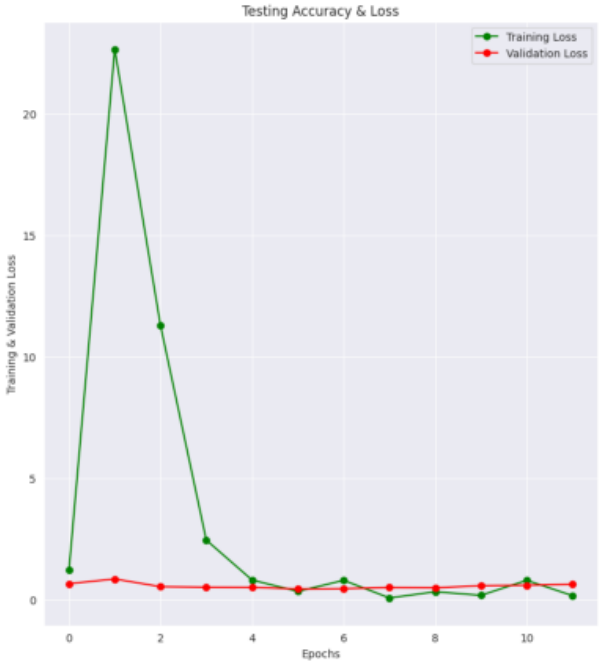

Fig. 17 Testing and accuracy loss

The training loss is getting in the decrease ratio and it reaches the good saturation point. Thus the genetic disease identification accuracy is calculated to make the prediction quality good.

VIII. CONCLUSION

A cutting-edge framework for detecting genetic diseases has been developed using deep neural networks and diverse medical data. The framework employs all X-ray images with genetic information for model training and data classification. By constructing functional intellectual networks based on signal correlation, the neural network formation is optimized using correlation coefficient information. This methodology greatly enhances diagnostic accuracy compared to traditional approaches, demonstrating that integrating advanced deep learning with medical expertise is an effective way to diagnose neurological disorders in their early stages. The same or similar methodologies can be applied to diagnose other neurological diseases, providing a foundation for ongoing diagnosis in this field. Assessing the effectiveness of deep learning techniques and algorithms for forecasting genetic disorders and their subcategories can enhance the model's precision. Additionally, scrutinizing the dataset has enabled us to identify the most suitable feature sets for model fitting. Health departments, clinics, and hospitals can utilize the model for real medical diagnosis, while the study's findings may prove valuable in genetic disorder lab experiments. Enhancing accessibility and usability can be achieved through the addition of a Graphical Interface, such as website applications. The application of disease predictive models in varied clinical populations can enhance the performance and limitations of the proposed models, thus refining medical practice. Though prediction remains a challenge, future research is promising and may provide a wealth of clinically useful information if evaluated within the appropriate context.

REFERENCES

[1] Nour eldeen m. khalifa 1, mohamed hamed n. taha 1, dalia ezzat ali 1, adam slowik 2, (senior member, ieee), and aboul ella hassanien "Artificial Intelligence Technique for Gene Expression by Tumor RNA-Seq Data: A Novel Optimized Deep Learning Approach". February 6, 2020.Digital Object Identifier 10.1109/IEEE ACCESS.2020.2970210

[2] Xiangxiang Zeng, Senior Member, IEEE, Yinglai Lin, Yuying He, Linyuan L¨u, Xiaoping Min∗, and Alfonso Rodr´ıguez-Pat´on" Deep collaborative filtering for prediction of disease genes". DOI 10.1109/TCBB.2019.2907536, IEEE/ACM Transactions on Computational Biology and Bioinformatics.

[3] W. R. J. Taylor and N. J. White, "Antimalarial drug toxicity: a review," Drug Saf., vol. 27, no. 1, pp. 25–61, 2004, doi: 10.2165/00002018200427010-00003.

[4] E. A. Ashley et al., "Spread of artemisinin resistance in Plasmodium falciparum malaria," N. Engl. J. Med., vol. 371, no. 5, pp. 411–423, Jul. 2014, doi: 10.1056/NEJMoa1314981.

[5] E. Tjitra et al., "Multidrug-resistant Plasmodium vivax associated with severe and fatal malaria: a prospective study in Papua, Indonesia,". PLoS Med., vol. 5, no. 6, p. e128, Jun. 2008, doi: 10.1371/journal.pmed.0050128.

[6] A. M. Dondorp et al., "Artemisinin Resistance in Plasmodium falciparum Malaria," N. Engl. J. Med., vol. 361, no. 5, pp. 455–467, Jul. 2009, doi: 10.1056/NEJMoa0808859.

[7] W. O. Godtfredsen, W. von Daehne, L. Tybring, and S. Vangedal, "Fusidic Acid Derivatives. I. Relationship between Structure and Antibacterial Activity," J. Med. Chem., vol. 9, no. 1, pp. 15–22, Jan. 1966, doi: 10.1021/jm00319a004.

[8] G. Kaur et al., "Synthesis of fusidic acid bioisosteres as antiplasmodial agents and molecular docking studies in the binding site of elongation factor-G," MedChemComm, vol. 6, no. 11, pp. 2023–2028, 2015, doi: 10.1039/C5MD00343A.

[9] S. Tonmunphean, V. Parasuk, and S. Kokpol, "QSAR Study of Antimalarial Activities and Artemisinin-Heme Binding Properties Obtained from Docking Calculations," Quant. Struct.-Act. Relatsh., vol. 19, no. 5, pp. 475–483, 2000, doi: 10.1002/15213838(200012)19:5<475::AID-QSAR475>3.0.CO;2-3.

[10] A. Worachartcheewan, C. Nantasenamat, C. Isarankura-Na-Ayudhya, and V. Prachayasittikul, "QSAR study of amidino bis-benzimidazole derivatives as potent anti-malarial agents against Plasmodium falciparum," Chem. Pap., vol. 67, no. 11, pp. 1462–1473, Nov. 2013, doi: 10.2478/s11696-013-0398-5.

[11] M. C. Sharma, S. Sharma, P. Sharma, and A. Kumar, "Pharmacophore and QSAR modeling of some structurally diverse azaaurones derivatives as anti-malarial activity," Med. Chem. Res., vol. 23, no. 1, pp. 181–198, Jan. 2014, doi: 10.1007/s00044-013-0609-1.

[12] M. Fernandez, J. Caballero, L. Fernandez, and A. Sarai, "Genetic algorithm optimization in drug design QSAR: Bayesian-regularized genetic neural networks (BRGNN) and genetic algorithm-optimized support vectors machines (GA-SVM)," Mol. Divers., vol. 15, no. 1, pp. 269–289, Feb. 2011, doi: 10.1007/s11030-010-9234-9. [

[13] https://colab.research.google.com/

[14] https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm

[15] https://www.codingforentrepreneurs.com/courses/python-google-colab-sheets-drive/

[16] J. T. Eppig, J. A. Blake, C. J. Bult, J. A. Kadin, and J. E. Richardson, "The mouse genome database (mgd): new features facilitating a model system," Nucleic Acids Research, vol. 35, no. Database issue, pp. 630–7, 2007.

[17] S. S. Dwight, M. A. Harris, K. Dolinski, C. A. Ball, G. Binkley, K. R. Christie, D. G. Fisk, L. Issel-Tarver, M. Schroeder, and G. Sherlock, "Saccharomyces genome database (sgd) provides secondary gene annotation using the gene ontology (go)," Nucleic Acids Research, vol. 30, no. 1, pp. 69–72, 2002.

[18]] T. L. Saito, M. Ohtani, H. Sawai, F. Sano, A. Saka, D. Watanabe, M. Yukawa, Y. Ohya, and S. Morishita, "Scmd: Saccharomyces cerevisiaemorphologicaldatabase,"NucleicAcidsResearch,vol.32, no. 1, pp. 319–22, 2004.

[19] K. L. Mcgary, I. Lee, and E. M. Marcotte, "Broad network-based predictability of saccharomyces cerevisiae gene loss-of-function phenotypes." Genome Biology, vol. 8, no. 12, p. R258, 2007.

[20] M. E. Hillenmeyer, E. Fung, J. Wildenhain, S. E. Pierce, S. Hoon, W. Lee, M. Proctor, R. P. St Onge, M. Tyers, and D. Koller, "The chemical genomic portrait of yeast: uncovering a phenotype for all genes." Science, vol. 320, no. 5874, pp. 362–365, 2008.