

“Early-Stage Disease Diagnosis System Using Human Nail Image Processing Using Deep Learning.”

1. INTRODUCTION :

1.1 Overview :

In the healthcare domain, many diseases can be predicted by observing the color and shape of human nails. A white spot here, a rosy stain there, or some winkler projection may be an indication of disease in the body. Problems in the liver, lungs, and heart can show up in your nails. Doctors observe the nails of patients to get assistance in disease identification. Usually, pink nails indicate a healthy human. Healthy nails are smooth and consistent in color. Anything else affecting the growth and appearance of the fingernails or toenails may indicate an abnormality. A person's nails can say a lot about their health condition. The need of such systems to analyze nails for disease prediction is because the human eye is having subjectivity about colors, having limitation of resolution and small amount of color change in a few pixels on the nail not being highlighted to human eyes which may lead to wrong result, whereas computer recognize small color changes on nails.

1.2 Purpose:

The project aim is to build a model which is used for the prevention and early detection of Nail Disease, Basically nail disease diagnosis depends on the different characteristics like color, shape, texture etc. Here the person can capture the images of the nail and then the image will be sent to the trained model. The model analyzes the image and detects whether the person is having nail disease or not and its type.

2 LITERATURE SURVEY:

2.1 Existing problem:

Some of the existing solutions for solving this problem are:

There are some existing systems which use color change of human palm for predicting disease. The system presented by Pandit et al. analyzes certain features in image and predicts probable disease using knowledge base of medical palmistry. Science of observing nails and palms to predict some diseases called Medical palmistry. Using example of the medical palmistry this model is discussed and presented for extraction of an interested portion of an image for further processing. This system presents work to get color of palm from the palm image and it works successfully on the different skin tones of human palm and increases accuracy of such observations of palm. The borders of palm are darker color than palm color in the scanned image of palm. Therefore exact boundary of palm may not found which can affect in the calculation of average color of palm. Image enhancement methods were used to overcome those problems. But this method is time consuming so to analyze nails it will take more time. So our system is using digital camera or mobile camera to get good quality image therefore it is easy to extract features of nails.

There is another model, for nail color analysis which predicts diseases using digital image processing. The system presented by Hardik Pandit et. al observes the color of nails using the principles of medical science as basis and output of system is prediction of diseases, if found any. In healthcare domain doctors often observe human nails to as supporting information or symptoms in disease prediction. The same task is defined by the proposed model without any human intervention. The model gives more accurate results than human vision, because it overcomes the limitations of human eye like subjectivity and resolution power.

As mentioned earlier different colors of nails indicate certain diseases. To implement this model, authors used computer based reference color values of ill nails to compare user's nail. That is, color values of nails of user's input images would be compared with these reference colors. If the case matches with any of reference color values, user would be victim of that disease. There will be 50 samples per color are taken for reference color values of ill nails. For example, for yellow nails, 50 different yellow nails color values are considered, and then their arithmetic mean is considered as a reference color value for yellow nails to RGB color components of reference color for that disease.

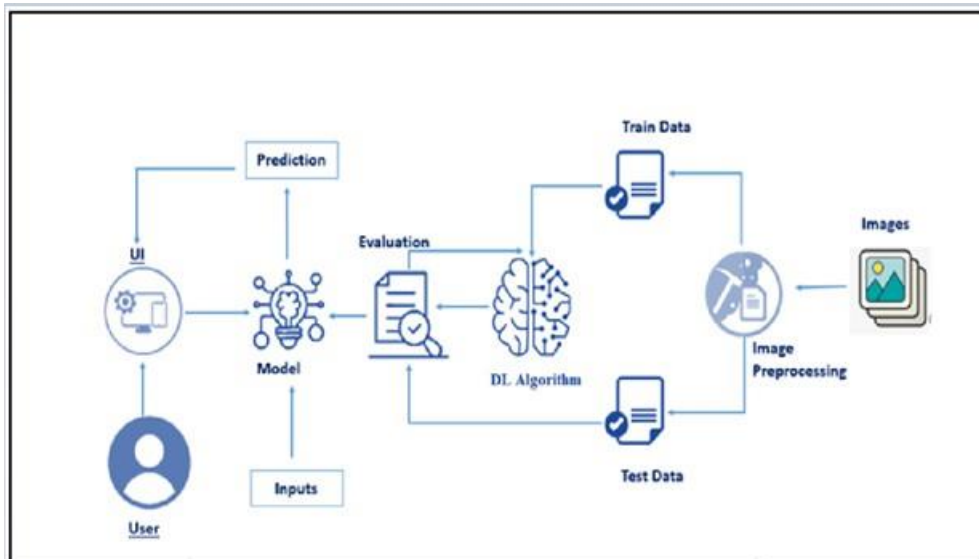
2.2 Proposed solution:

There are different ways of disease diagnosis such as through various tests (blood test, urine test etc.) and symptoms available on various parts of body guides towards disease diagnosis. We are proposing a system which will take nail image as an input and output will be possible diseases prediction based on color changes. This is our first step towards a perfect system so we are not keen to accuracy of system. The main objective of this system design is to provide an application for use in healthcare domain this is advantageous in terms of cost and time. The proposed system will take nail image as an input and will perform some processing on input image. Then finally it will predict probable diseases. This system can be used by people as well as by doctors in healthcare domain

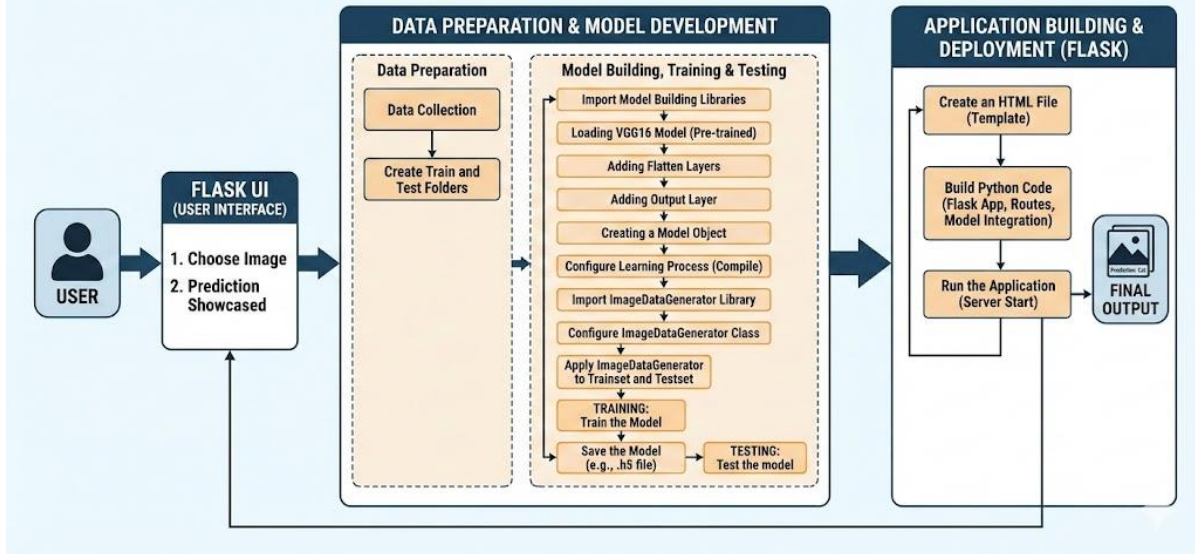
3 THEORITICAL ANALYSIS:

3.1 Block Diagram:

Architecture:



PROJECT FLOW: VGG16 IMAGE CLASSIFICATION WITH FLASK UI



3.2 Hardware / Software designing:

Hardware Requirements:

Operating System	Windows, Mac, Linux
CPU (for training)	Multi Core Processors (i3 or above/equivalent)
GPU (for training)	NVIDIA AI Capable / Google's TPU

Software Requirements:

Python	v3.10.0 or Above
Python Packages	flask, tensorflow, opencv-python, keras, numpy, pandas, scikit-learn.
Web Browser	Mozilla Firefox, Google Chrome or any modern web browser

3. EXPERIMENTAL INVESTIGATIONS:

The performance of the CNNs trained with the nail dataset was estimated by the classification performance of the models with the validation datasets. The performance of fine image selector helps in assessing image quality with the change in the illumination and reduction noise level of the images. The levels of brightness and noise were gradually reduced to classify the image easily. If the validation loss decreases then the accuracy will increase. The number of epochs should be as high as possible and terminate training based on the error rates. An epoch is one learning cycle where the learner sees the whole training data set. Here we are having 100 epochs and we are getting 100% validation and training accuracy. In this experiment we found that using colour feature of nail image average 100% results are correctly matched with training set data during three tests conducted and we are getting 100% accuracy on test dataset.

4. FLOWCHART:

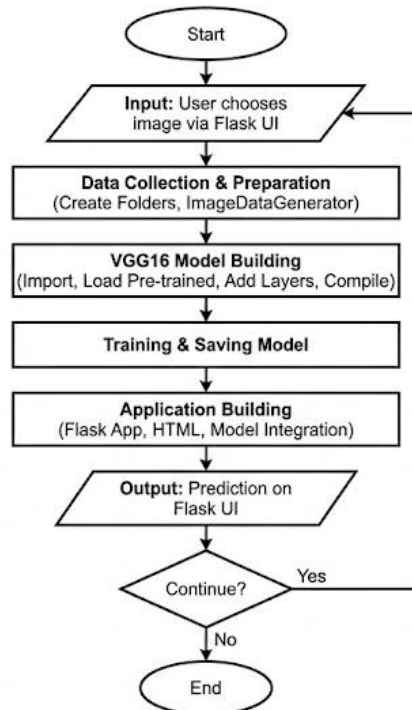


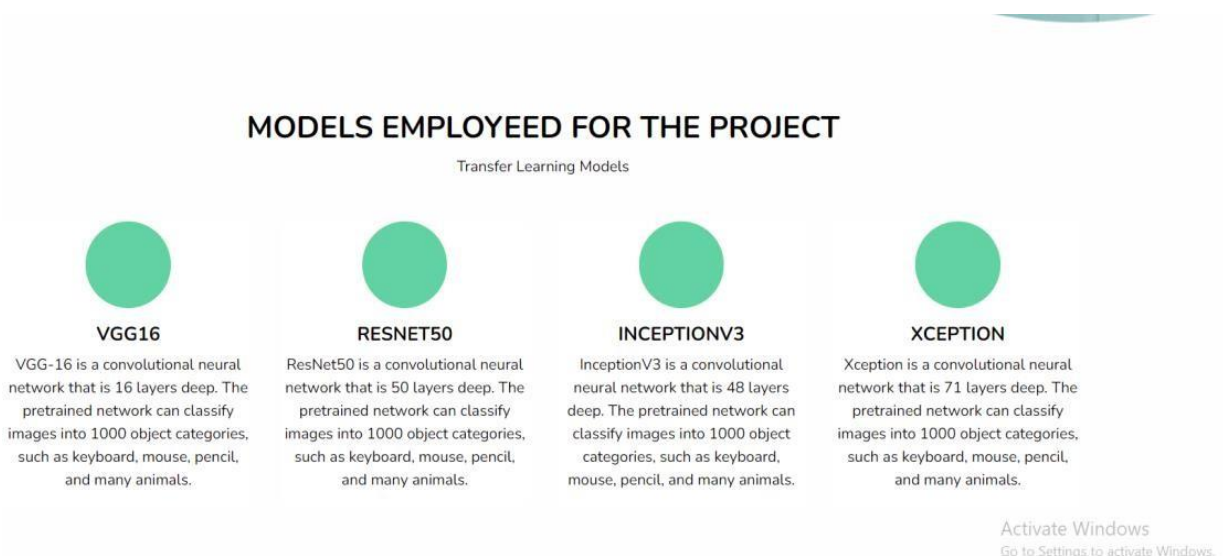
Figure 3. Flowchart of VGG16 Image Classification Project with Flask UI.

5. RESULT:

The proposed procedure was implemented and tested with set of images. The set of 874 images of different nail images are used for training the database and all the images are used for testing the database. Once the image is given to the model the disease will be shown on the screen.



Information about VGG16 Regarding:





ABOUT US

Health systems are under greater pressure than ever, but along with its other uses, the potential of AI to provide much-needed relief to overworked doctors. can streamline the triage process. It seamlessly integrates into the medical workflow and uses the same logic physicians embrace at medical school, delivering enhanced insights by calculating the likelihood of particular conditions based on the patient's symptoms.AI can facilitate a faster, automated route to outcome by reducing the steps that doctors need to take, ultimately meaning quicker answers and patient recovery.AI can help providers determine who really needs to see a doctor.

Main Project Flast UI:

OUR PROJECT ON NAIL DISEASE

Human nail can be used for the prediction of various systemic and dermatological diseases. The proposed system – Nail Image Processing System helps us to the analysis of human nail and thereby help us in predicting various nail diseases. The input to the proposed system is the Human Palm Image. The nail portion is segmented and nail color, shape and texture features are extracted and taken together to form a feature vector and then analysis of nail is done which is used for the diagnosis of various nail diseases.is proposed system will help the doctors in the early diagnosis of diseases.

Predict

Reach at..

📍 Location
☎ Call +01 1234567890
✉ demo@gmail.com



About

Health systems are under greater pressure than ever, but along with its other uses, the potential of AI to provide much-needed relief to overworked doctors. can streamline the triage process.

Links

[Home](#) [About](#) [Nail](#)

Newsletter

Enter email

Subscribe

Windows
Go to Settings to activate Windows.

After Prediction UI with Result:

The screenshot displays the 'NAIL DISEASE PREDICTION' web application. At the top, a green header contains the title and navigation links: HOME, ABOUT, NAIL, and a search icon. Below the header, a file upload section includes a 'Choose File' button, the text 'No file chosen', and a 'Submit' button. The main content area shows the prediction result: 'The Person is diagnosed with Nail Disease'. At the bottom, a green footer contains four columns: 'Reach at..' with contact information (Location, Call: +01 1234567890, Email: demo@gmail.com, and social media icons for Facebook, Twitter, LinkedIn, and Instagram); 'About' with a paragraph about AI in healthcare; 'Links' with 'Home', 'About', and 'Nail' links; and 'Newsletter' with an email input field and a 'Subscribe' button. A small 'Activate Windows' watermark is visible in the bottom right corner.

6. ADVANTAGES AND DISADVANTAGES:

Advantages:

1. It is possible to create a mobile application to let the people known about the nails and detect the disease in an early stage.
2. As different types of nail images are present in the dataset more types of nail images can be added and user can choose the specific image to diagnose the disease.

Disadvantages:

1. The current model only works with the images present on the dataset.
2. There is no option for the user to take a picture of their nail and get the disease diagnosis.

7. APPLICATIONS:

1. It will contribute for the early disease detection via human nail image processing.
2. As a result , people can identify their health status, resolve health issues and safeguard them accordingly with web app .

8. CONCLUSION:

There are five main parts of nail namely lunula, cuticle, nail root, nail plate, nail lines out of which we used nail plate is used in ESDD system. In presented system, system analyzes the human nail and gives probable disease for person including healthy case. Here, for disease prediction nail color (average RGB) value used as a nail feature. This model gives more accurate results than human vision, because it overcomes the limitations of human eye like subjectivity and resolution power.

9. FUTURE SCOPE:

- The trained Model can be compressed for Edge Device using different architecture like TensorFlow Lite. (TFLite).
- Their results and performance be measured for publication
- Each of Disorder can be trained using different Transfer Learning Architectures and their results can be compared for publication.
- A Smart Application can be developed after compressing the trained model.
- A device can also be designed and developed using Embedded Vision and it should be patented with International Scope

10. BIBLIOGRAPHY

1. Environment Setup:

- <https://www.youtube.com/watch?v=5mDYijMfSzs>

2. Dataset :

- <https://drive.google.com/drive/folders/1AXTYsbiarS1TCAgfj0mancTSrJYYMWMS>

3. Keras Image Processing Doc:

- <https://keras.io/api/preprocessing/image/>

4. Keras Image Dataset From Directory Doc:

- <https://keras.io/api/preprocessing/image/#imagedatasetfromdirectory-function>

5. Flask Basics:

- https://www.youtube.com/watch?v=Ij4I_CvBnt0

6. IBM Academic Partner Account Creation:

- <https://www.youtube.com/watch?v=x6i43M7BAqE>

APPENDIX:

Training and testing the dataset:

```
import sys
# fit the model
r = model.fit_generator(
    training_set,
    validation_data=test_set,
    epochs=100,
    steps_per_epoch=len(training_set)//3,
    validation_steps=len(test_set)//3)
```

```
Epoch 85/100
7/7 [=====] - 10s 1s/step - loss: 0.4130 - accuracy: 0.9107 - val_loss: 0.2166 - val_accuracy: 0.9844
Epoch 86/100
7/7 [=====] - 10s 1s/step - loss: 0.4140 - accuracy: 0.9286 - val_loss: 0.2503 - val_accuracy: 0.9531
Epoch 87/100
7/7 [=====] - 9s 1s/step - loss: 0.4447 - accuracy: 0.9062 - val_loss: 0.2644 - val_accuracy: 0.9844
Epoch 88/100
7/7 [=====] - 9s 1s/step - loss: 0.4943 - accuracy: 0.8889 - val_loss: 0.2438 - val_accuracy: 0.9688
Epoch 89/100
7/7 [=====] - 9s 1s/step - loss: 0.3800 - accuracy: 0.9241 - val_loss: 0.3644 - val_accuracy: 0.9219
Epoch 90/100
7/7 [=====] - 10s 1s/step - loss: 0.3542 - accuracy: 0.9330 - val_loss: 0.2475 - val_accuracy: 0.9375
Epoch 91/100
7/7 [=====] - 9s 1s/step - loss: 0.3644 - accuracy: 0.9324 - val_loss: 0.2120 - val_accuracy: 0.9844
Epoch 92/100
7/7 [=====] - 9s 1s/step - loss: 0.4158 - accuracy: 0.9082 - val_loss: 0.1973 - val_accuracy: 1.0000
Epoch 93/100
7/7 [=====] - 9s 1s/step - loss: 0.4182 - accuracy: 0.9107 - val_loss: 0.2503 - val_accuracy: 0.9688
Epoch 94/100
7/7 [=====] - 9s 1s/step - loss: 0.3971 - accuracy: 0.9196 - val_loss: 0.3026 - val_accuracy: 0.9531
Epoch 95/100
7/7 [=====] - 10s 1s/step - loss: 0.4171 - accuracy: 0.9107 - val_loss: 0.2758 - val_accuracy: 0.9688
Epoch 96/100
7/7 [=====] - 9s 1s/step - loss: 0.3769 - accuracy: 0.9227 - val_loss: 0.1795 - val_accuracy: 0.9844
Epoch 97/100
7/7 [=====] - 9s 1s/step - loss: 0.3770 - accuracy: 0.9330 - val_loss: 0.2502 - val_accuracy: 0.9688
Epoch 98/100
7/7 [=====] - 9s 1s/step - loss: 0.3488 - accuracy: 0.9375 - val_loss: 0.2567 - val_accuracy: 0.9688
Epoch 99/100
7/7 [=====] - 9s 1s/step - loss: 0.3582 - accuracy: 0.9082 - val_loss: 0.3165 - val_accuracy: 0.9375
Epoch 100/100
7/7 [=====] - 9s 1s/step - loss: 0.3040 - accuracy: 0.9469 - val_loss: 0.2189 - val_accuracy: 0.9844
```

```
#save the model
model.save('vgg-16-nail-disease.h5')
```

```
#import load_model class for loading h5 file
from tensorflow.keras.models import load_model
#import image class to process the images
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import numpy as np
```

```
#load saved model file
model=load_model('vgg-16-nail-disease.h5')
```

```
#load one random image from local system
img=image.load_img(r'19.PNG',target_size=(224,224))
```

```
#convert image to array format
x=image.img_to_array(img)
```

```
model.predict(img_data)
```

```
array([[3.7915495e-06, 2.1875452e-03, 4.1685593e-07, 4.1365284e-03,
        5.5700364e-11, 7.3820658e-08, 5.7155389e-06, 1.0331855e-06,
        6.6932332e-04, 9.1808915e-01, 4.5551822e-08, 1.2603366e-03,
        2.2272835e-12, 7.7233365e-04, 7.8795410e-06, 6.9824591e-02,
        3.0412099e-03]], dtype=float32)
```

```
output=np.argmax(model.predict(img_data), axis=1)
```

```
index=['Darier_s disease', 'Muehrck-e_s lines', 'alopecia areata', 'beau_s lines', 'bluish nail',
        'clubbing','eczema','half and half nailes (Lindsay_s nails)','koilonychia','leukonychia',
        'onycholycis','pale nail','red lunula','splinter hemorrhage','terry_s nail','white nail','yellow nails']
result = str(index[output[0]])
result
```

```
'leukonychia'
```

Web Application in Flask:

```
1  #import re
2  import numpy as np
3  import os
4  from flask import Flask, app,request,render_template
5  from tensorflow.keras import models
6  from tensorflow.keras.models import load_model
7  from tensorflow.keras.preprocessing import image
8  from tensorflow.python.ops.gen_array_ops import concat
9  from tensorflow.keras.applications.inception_v3 import preprocess_input
10 import requests
11 from flask import Flask, request, render_template, redirect, url_for
```

```
#default home page or route
@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/about')
def about():
    return render_template("about.html")
```

```
@app.route('/nailpred')
def nailpred():
    return render_template('nailpred.html')
```

```
@app.route('/nailhome')
def nailhome():
    return render_template('nailhome.html')
```

```

@app.route('/nailresult',methods=["GET","POST"])
def nres():
    if request.method=="POST":
        f=request.files['image']
        basepath=os.path.dirname(__file__) #getting the current path i.e where app.py is present
        #print("current path",basepath)
        filepath=os.path.join(basepath,'uploads',f.filename) #from anywhere in the system we can give image but we want the
        #print("upload folder is",filepath)
        f.save(filepath)

        img=image.load_img(filepath,target_size=(224,224))
        x=image.img_to_array(img)#img to array
        x=np.expand_dims(x,axis=0)#used for adding one more dimension
        #print(x)
        img_data=preprocess_input(x)
        prediction=np.argmax(modeln.predict(img_data))

        index=['Darier_s disease', 'Muehrck-e_s lines', 'alopecia areata', 'beau_s lines', 'bluish nail',
               'clubbing','eczema','half and half nailes (Lindsay_s nails)','koilonychia','leukonychia',
               'onycholycis','pale nail','red lunula','splinter hemmorrhage','terry_s nail','white nail','yellow nails']
        nresult = str(index[prediction])
        nresult

```

```

""" Running our application """
if __name__ == "__main__":
    app.run(debug =False, port = 8080)

```