YOGA POSE CLASSIFICATION USING DEEP LEARNING WITH IBM WATSON

1.INTRODUCTION

1.1. OVERVIEW

Nowadays, yoga has gained worldwide attention due to increased stress levels in the modern lifestyle, and there are numerous methods or resources for learning yoga. Yoga can be practiced in yoga centers, through personal tutors, and can also be learned on one‟s ownwith the help of the Internet, books, recorded clips, etc.In fast-paced lifestyles, many people prefer self learning because the above mentioned resources might not be available all the time. But in self learning, one may not find an incorrect pose.

1.2. PURPOSE :

The problem with yoga however is that, just like any other exercise, it is of tmost importance to practice it correctly as any incorrect posture during a yoga session can be unproductive and possibly detrimental. This leads to the necessity of having an instructor to supervise the session and correct the individual's posture. Since not all users have access or resources to an instructor,artificial intelligence-based application might be used to identify yoga poses and provide personalized feedback to help individuals improve their form.

2.LITERATURE SURVEY

INTRODUCTION

Yoga is a 5000 year old practice developed in ancient India by the indus-sarasvati Civilization.The word yoga means deep association and union of mind with the body.It is used to keep both mind and body in equilibration in all flip-flops of the lives. Now a days yoga has gained worldwide attention due to increased stress levels in themodern lifestyle.Many people prefer self learning because the above mentioned resources might not be available all the time.But in self-learning one may not find an incorrect pose.Incorrect posture can be harmful to one's health,resulting in acute pain and longterm chronic concerns.In this project deep learning based techniques are developed to detect yoga pose by uploading an image or by real time video using computer vision technique.It detecys the pose and tells exactly the pose name.Transfer learning has become one of the most common techniques that has achieved better performance in many areas,especially in medical image analysis and classification.We used transfer learning techniques like Inception V3,ResNet-50,Xception,VGG19 that are more widely used as a transfer learning methos in medical image analysis and they are highly effective.

2.1.EXISTING PROBLEM

The purpose of the present model is to predict the whether a person is practicing correct yoga pose or not. The model examines data from users concenns. Here we are building a model by applying various machine learning algorithms find the best accurate model. And integrate to flask based web application. User can predict the disease by entering parameters in the web application.

2.2. PROPOSED SOLUTION :

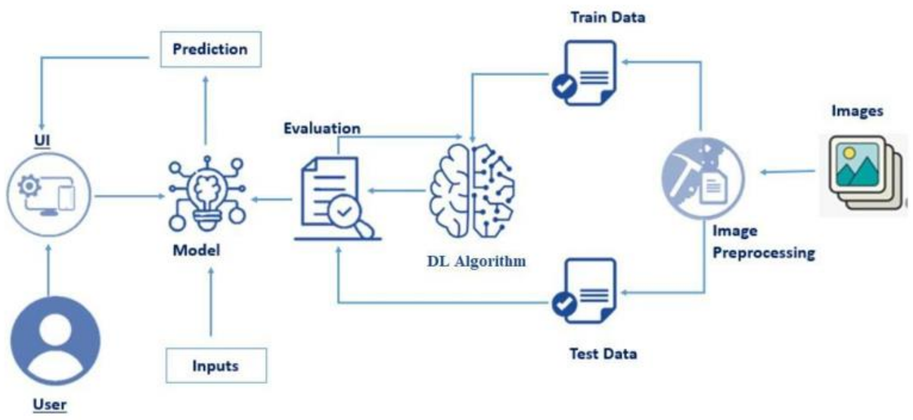
Here we are building a model by applying various machine learning algorithms find the best accurate model. Thus a person will get to know whether he/she is posing correctly or not.

Some of the machines learning algorithms are:

- Inception v3
- ResNet50
- Xception
- VGG19

3.THEORETICAL ANALYSIS :

3.1 BLOCK DAIGRAM



3.2 HARDWARE OR SOFTWARE DESIGN

1.Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum, and first released on February 20, 1991. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

2.Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, RStudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

3.Jupyter Notebook

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an I Python Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

4.Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Spyder is extensible with first-party and third-party plugins includes support for interactive tools for data inspection and embeds Python specific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

5.Flask

Web framework used for building. It is a web application framework written in python which will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the user.

6.TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning with a particular focus on deep neural networks. Deep learning is a subset of machine learning that involves the analysis of large-scale unstructured data. Deep learning differs from traditional machine learning in that the latter typically deals with structured data. TensorFlow boasts of a flexible and comprehensive collection of libraries, tools, and community resources.

7.Keras

Karas runs on top of open-source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks.

8.Layersopen cv

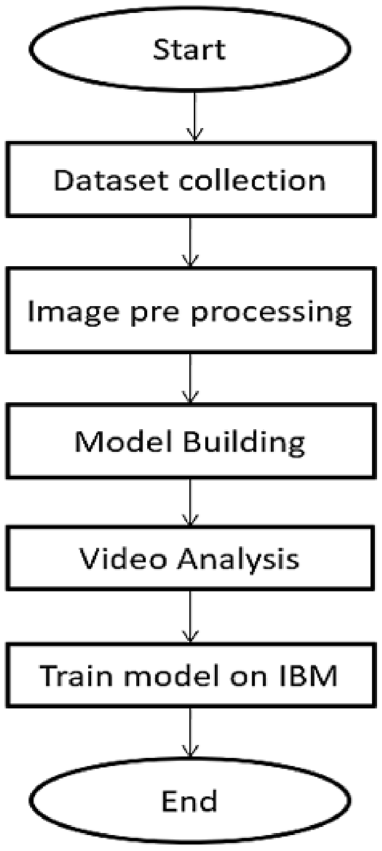
OpenCV is created to implement various operations including recognizing and detecting faces, analysing human tasks in videos, identifying objects, recording camera movements, tracking moving objects, and combining images to create a high-resolution image for the accurate scene. Let's see the topic defining the term "Computer Vision."

Hardware Requirements:

1.	Operating System	:	windows 7 and above with 64 bits
2.	Processor Type	:	Intel Core i3-3220 and above
3.	RAM	:	4Gb and above
4.	Hard Disk	:	minimum 100GB

4. EXPERIMENTAL INVESTIGATION

The text data need to be organized before proceeding with the project. The original dataset has a single folder. We will be using the HDI.csv file to fetch the text data of training data. The data need to be unique and all fields need to be filled. The dataset images are to be pre-processed before giving to the model. We will create a function that uses the pre-trained model for predicting custom outputs. Then we have to test and train the model. After the model is build, we

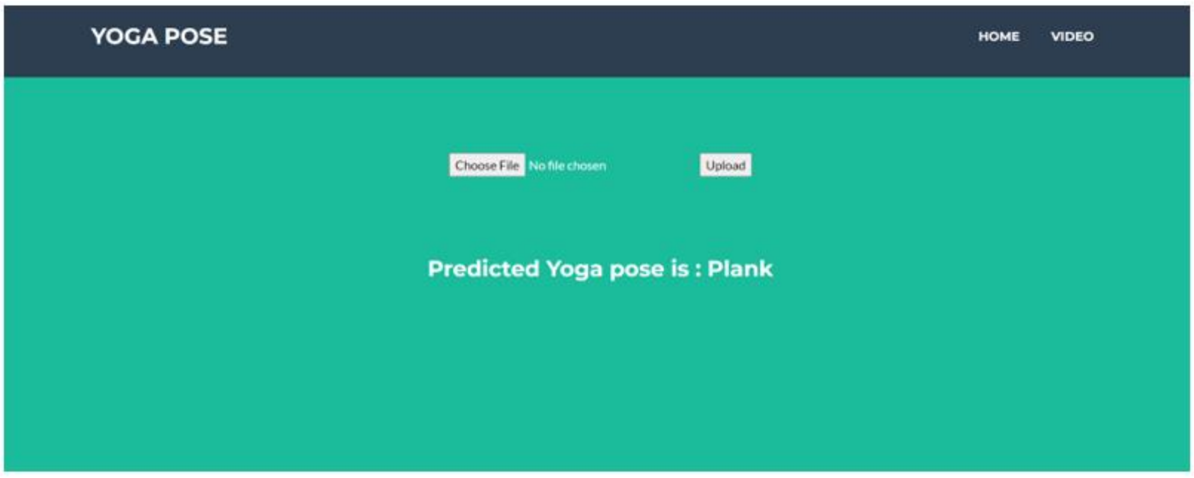


will be integrating it to a web application.

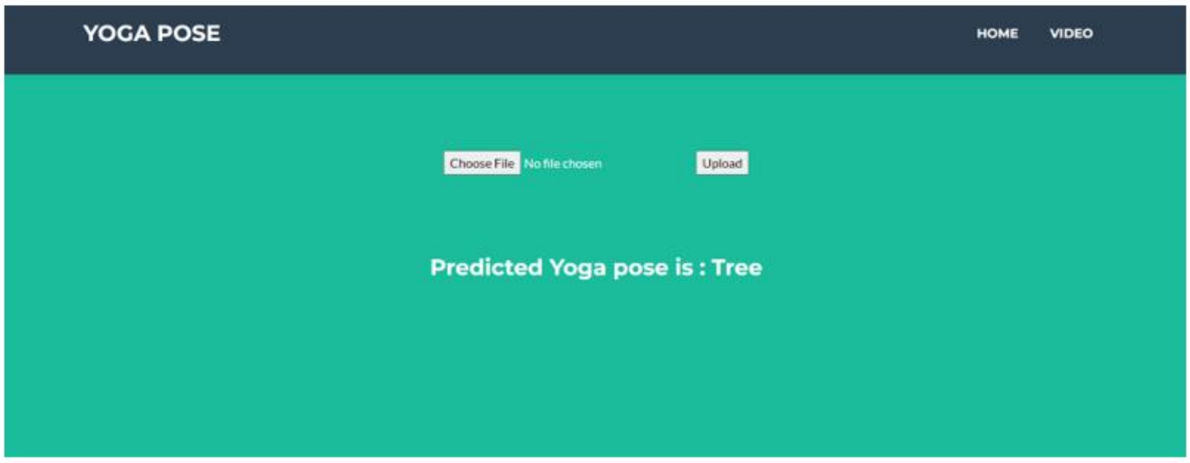
5. FLOWCHART

6.RESULT

As the first set data set is collected.We know that image which is uploaded contains the yoga pose.So the system should detect the correct yoga pose . This is performed for each image.



Predicted Yoga pose is : Plank



Predicted Yoga pose is : Tree

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

1. Features are automatically deduced and optimally tuned for desired outcome. Features are not required to be extracted ahead of time.

2. The same neural network-based approach can be applied to many different application and data type.

DISADVANTAGES

3. Screen is difficult to read in low light or bright sunlight
4. Keeping track of cameras and images (no meta-data)
5. Losing settings during transit
6. Walk-by test requires downloading image on laptop in the field.

8. APPLICATIONS

1. Object Classification and Detection in Photographs
2. Detecting exact yoga postures
3. Automatic Handwriting Generation
4. Character Text Generation
5. Automatic Machine Translation
6. Image Caption Generation

9. CONCLUSION

The paper discusses in detail all advances in the area of yoga pose classification. Yoga pose classification is very useful for many real time applications. This helps the people to practice yoga in a very effective manner.And helps to know about all yoga postures and the correct way of doing yoga.This eliminates the painful experience faced by the people while practicing yoga.

10. FUTURE SCOPE

It must be noticed that our methodology annihilates the requirement for kinect or some other specific equipment for yoga pose identifictaion and can be actualized on contribution from an odinary RGB camera.In future work,more asanas and bigger dataset involving both picture and recordings can be incorporated**.**

11.BIBILOGRAPHY

1.www.google.com

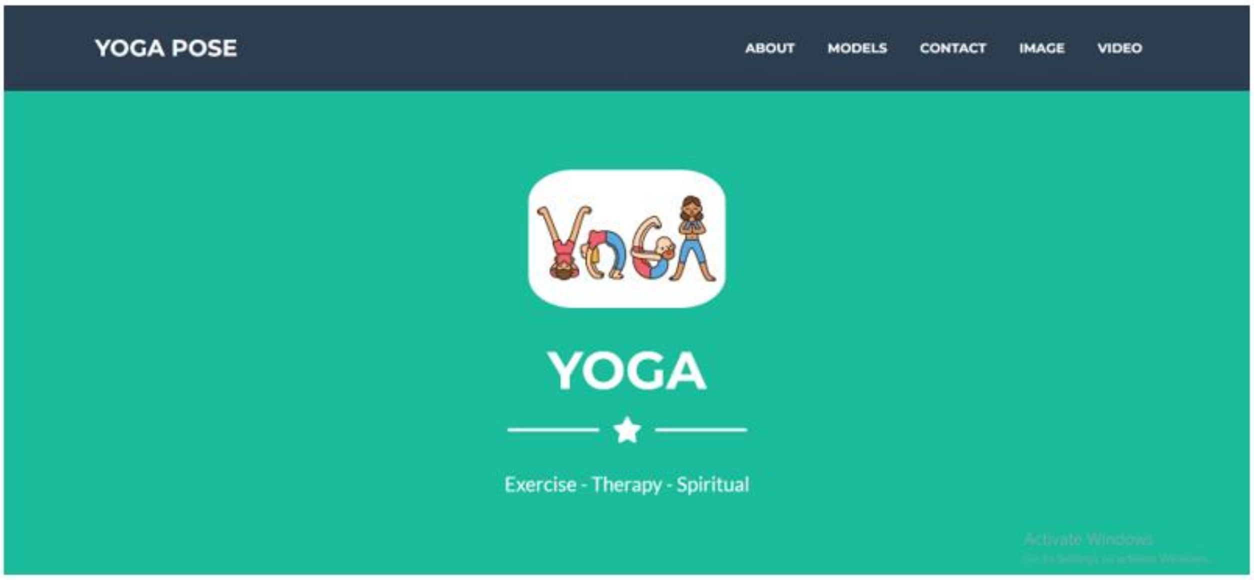
2.www.wikipedia.org

3.https://ieeexplore.ieee.org

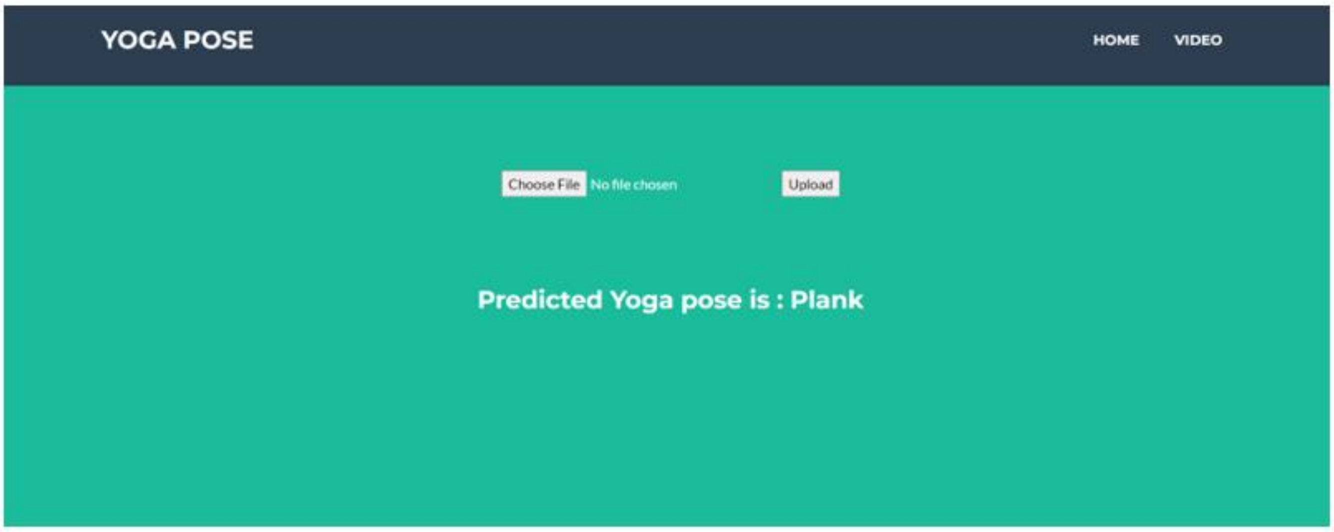
12 APPENDIX

```
from flask import Flask, request, render_template
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.xception import preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import load_img
import numpy as np
import os
import cv2
from werkzeug.utils import secure_filename
model = load_model("xcep_yoga.h5")
app = Flask(__name__)
@app.route("/")
def home():
return render_template("index.html")
@app.route("/home")
def homeagain():
return render_template("index.html")
@app.route("/iinput")
def predict():
return render_template("input.html")
```

```
@app.route("/ioutput", methods=['POST','GET'])
def output():
if request.method == 'POST':
img = request.files["file"]
img.save("uploaded_image.png")
img = load_img("uploaded_image.png", target_size=(224,224))
img = image.img_to_array(img)
img.shape
x = np.expand_dims(img, axis=0)
img_data=preprocess_input(x)
pred = model.predict(img_data)
p = np.argmax(pred)
columns = ['Downdog', 'Goddess', 'Plank', 'Tree', 'Warrior2']
result = str(columns[p])
return render_template("input.html", prediction = result)
@app.route("/voutput", methods=['POST','GET'])
def video():
# Get a reference to webcam #0 (the default one)
print("[INFO] starting video stream...")
vs = cv2.VideoCapture(0)
#writer = None
(W, H) = (None, None)
# loop over frames from the video file stream
while True:
# read the next frame from the file
(grabbed, frame) = vs.read()
# if the frame was not grabbed, then we have reached the end
# of the stream
if not grabbed:
break
# if the frame dimensions are empty, grab them
if W is None or H is None:
(H, W) = frame.shape[:2]
# clone the output frame, then convert it from BGR to RGB
# ordering and resize the frame to a fixed 64x64
output = frame.copy()
#print("apple")
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
frame = cv2.resize(frame, (224, 224))
#frame = frame.astype("float32")
x=np.expand_dims(frame, axis=0)
result = np.argmax(model.predict(x), axis=-1)
index=['Downdog', 'Goddess', 'Plank', 'Tree', 'Warrior2']
result=str(index[result[0]])
cv2.putText(output, "Pose: {}".format(result), (10, 120), cv2.FONT_HERSHEY_PLAIN,
1, (0,255,255), 1)
#playaudio("Emergency it is a disaster")
cv2.imshow("Output", output)
key = cv2.waitKey(1) & 0xFF
# if the q key was pressed, break from the loop
if key == ord("q"):
break
# release the file pointers
print("[INFO] cleaning up...")
vs.release()
cv2.destroyAllWindows()
return render_template("output.html")
```

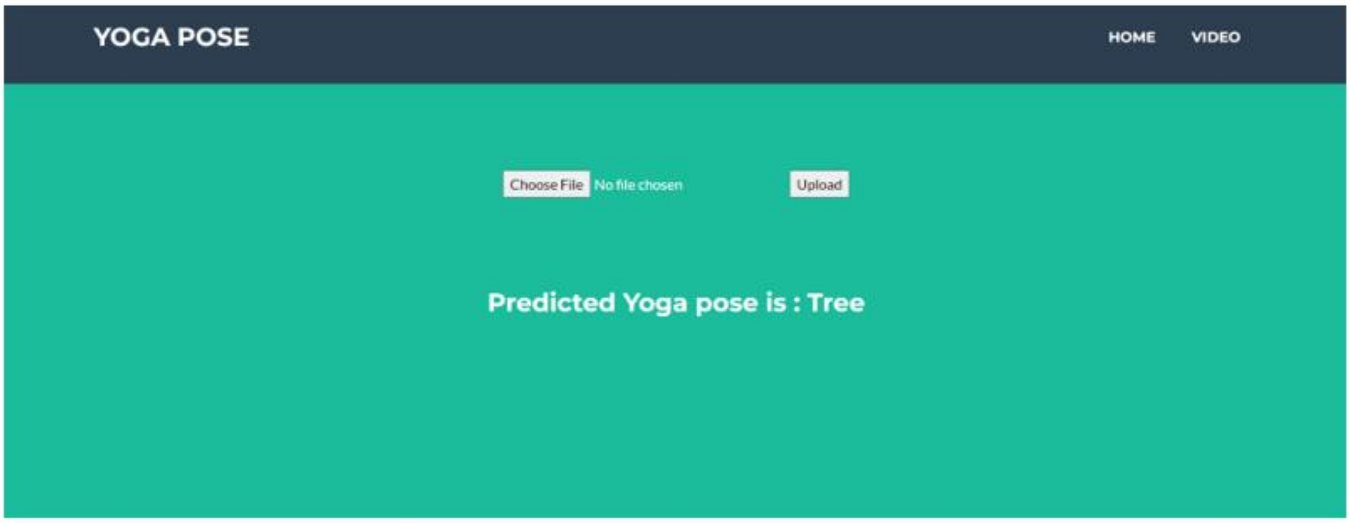


```
if name == '__main__':
app.run(debug=True)
```



Predicted Yoga pose is : Plank

OUTPUT



Predicted Yoga pose is : Tree