LANDMARK DETECTION

* **ABSTRACT:**

Landmark detection, a fundamental task in computer vision, involves identifying and localizing crucial points of interest within an image. This project aims to explore and implement landmark detection techniques using deep learning models, focusing on their application in facial keypoints localization. The objective is to develop an accurate and efficient system capable of detecting facial landmarks for use in various real-world applications, such as facial recognition, augmented reality, and facial expression analysis.

The project begins with an in-depth review of landmark detection algorithms, architectures, and methodologies that have gained prominence in the computer vision community. A comprehensive study is conducted to understand the strengths and limitations of various state-of-the-art models, such as Convolutional Neural Networks (CNNs), Hourglass networks, and Transformer-based architectures.

A diverse dataset containing annotated facial images with labeled keypoints is curated and preprocessed to ensure data quality and suitability for training. Data augmentation techniques are employed to augment the dataset, improving the model's generalization capabilities.

The selected deep learning model is trained using a combination of classification and regression losses, effectively learning to predict facial landmarks' coordinates. Hyper parameters are tuned using a validation set to achieve optimal performance while avoiding overfitting. The training process is performed on a GPU-equipped platform to expedite the training time.

The trained model's performance is rigorously evaluated on a separate validation set using various metrics, including Mean Euclidean Distance (MED) and Percentage of Correct Keypoints (PCK). A detailed analysis of the results is presented, highlighting the model's strengths, weaknesses, and potential areas for further improvement.

To enhance the model's accuracy, fine-tuning and optimization techniques are employed, aiming to achieve state-of-the-art landmark detection performance. The iterative process of refining the model is documented and discussed in-depth.

The final trained model is integrated into a real-time facial landmark detection system, demonstrating its effectiveness and efficiency in keypoint localization. The system is deployed on a chosen platform, showcasing its practical application in a user-friendly interface.

The completion of this project not only contributes to the understanding of landmark detection methodologies but also offers a valuable asset for various computer vision applications that rely on accurate keypoint localization. The work presented in this project serves as a foundation for future advancements and enhancements in the field of landmark detection using deep learning techniques.

* **OBJECTIVE**:

The primary objective of this landmark detection project is to explore, implement, and evaluate deep learning techniques for accurate and efficient landmark localization, with a specific focus on facial keypoints. The project aims to achieve the following key goals:

**Comprehensive Study:** Conduct an in-depth study of landmark detection algorithms, architectures, and methodologies, focusing on deep learning-based approaches. Analyze the strengths and limitations of state-of-the-art models to identify the most suitable techniques for facial keypoint localization.

**Dataset Curation:** Curate a diverse and annotated dataset containing facial images with labeled keypoints. Preprocess the dataset to ensure data quality, balance, and suitability for training the landmark detection model.

**Model Development:** Implement and train a deep learning model for landmark detection. Choose an appropriate architecture and loss functions to enable the model to learn and predict facial keypoints' coordinates accurately.

**Performance Evaluation:** Evaluate the trained model's performance using various metrics, such as Mean Euclidean Distance (MED) and Percentage of Correct Keypoints (PCK). Thoroughly analyze the results to assess the model's accuracy and robustness.

**Fine-tuning and Optimization:** Apply fine-tuning and optimization techniques to enhance the model's performance further. Experiment with hyperparameters to achieve state-of-the-art accuracy while avoiding overfitting.

**Real-time Landmark Detection:** Integrate the trained model into a real-time facial landmark detection system. Demonstrate the system's efficiency in localizing facial keypoints in live video or image streams.

**Contribution to the Field:** Contribute valuable insights and advancements to the field of landmark detection using deep learning. Provide a solid foundation for future research and development in accurate keypoint localization.

**Learning and Skill Development:** Gain practical experience in computer vision, deep learning, and project development. Improve technical skills in data preprocessing, model training, and performance analysis.

By achieving these objectives, the project aims to contribute to the growing field of computer vision and pave the way for more sophisticated and accurate landmark detection systems with practical applications in various domains.

* **INTRODUCTION:**

Landmark detection is a crucial computer vision task that involves the precise localization and identification of key points or landmarks within an image. These landmarks serve as essential reference points, playing a significant role in a wide range of applications, including facial recognition, pose estimation, medical imaging, and augmented reality. One particularly important area of landmark detection is the localization of facial keypoints, which has numerous practical applications in facial analysis and human-computer interaction.

This project aims to delve into the realm of landmark detection using deep learning techniques and focuses on the specific application of facial keypoint localization. By leveraging the advancements in deep learning, we aim to develop an accurate and efficient landmark detection system that can robustly identify and localize facial keypoints in real-time. The project's ultimate objective is to contribute to the field of computer vision by providing a comprehensive study, implementation, and evaluation of landmark detection methodologies.

The significance of facial keypoint localization lies in its vast potential for various real-world applications. In facial recognition systems, accurate localization of facial keypoints, such as the eyes, nose, and mouth, can lead to improved identification accuracy and robustness against variations in pose and lighting conditions. Facial keypoint localization is also instrumental in facial expression analysis, enabling the understanding of human emotions and facilitating applications like emotion recognition and virtual avatar animation.

To achieve this, we will start by studying different landmark detection algorithms and deep learning models. Then, we will create a dataset of facial images with labeled keypoints and preprocess the data for training. Using the dataset, we will train a deep learning model to predict the coordinates of facial keypoints.

We will evaluate the model's performance and fine-tune it for better accuracy. Once the model is ready, we will integrate it into a real-time facial landmark detection system. The system will demonstrate how effectively it can identify facial keypoints in live video or images.

Throughout the project, we will document our progress and insights to contribute to the field of computer vision. By the end, we aim to have a reliable and user-friendly system for accurate facial keypoint localization using deep learning techniques.

In conclusion, this landmark detection project embarks on a journey to leverage the power of deep learning to develop an accurate and efficient facial keypoint localization system. By accomplishing our objectives, we aspire to contribute to the field of computer vision and pave the way for more sophisticated and practical applications of landmark detection technology in various domains

* Top of Form
* **METHODOLOGY:**

1. **Literature Review:**

* Conduct a thorough review of landmark detection algorithms and deep learning models.
* Focus on studying methodologies for facial keypoint localization.
* Identify the most effective and widely used approaches in the field.

1. **Dataset Preparation:**

* Collect or obtain a dataset containing facial images with annotated keypoints.
* Preprocess the dataset by resizing images and normalizing pixel values.
* Augment the data with image transformations (e.g., rotation, flipping) to increase dataset diversity.

1. **Model Selection:**

* Choose a suitable deep learning model for facial landmark detection.
* Consider factors such as model complexity, computational requirements, and previous performance on similar tasks.

1. **Model Development:**

* Design the architecture of the chosen model, including input and output layers.
* Implement the model using a deep learning framework (e.g., TensorFlow or PyTorch).
* Define the loss function and optimization algorithm for training the model.

1. **Model Training:**

* Split the dataset into training and validation sets.
* Train the model on the training set using backpropagation and gradient descent.
* Monitor the model's performance on the validation set to avoid overfitting.

1. **Model Evaluation:**

* Evaluate the trained model's performance using evaluation metrics like Mean Euclidean Distance (MED) and Percentage of Correct Keypoints (PCK).
* Analyze the results to understand the model's accuracy and potential areas for improvement.

1. **Fine-tuning and Optimization:**

* Fine-tune the model by adjusting hyperparameters and regularization techniques.
* Optimize the model for better accuracy and generalization.

1. **Real-time Implementation:**

* Integrate the trained model into a real-time facial landmark detection system.
* Utilize appropriate libraries and tools to process live video or image streams.

1. **System Testing:**

* Test the real-time system with diverse facial images and videos.
* Assess the system's performance in terms of speed and accuracy.

By following this methodology, we aim to develop a reliable and efficient facial landmark detection system using deep learning. The step-by-step approach ensures a well-structured project that facilitates successful implementation and achievement of the project's objectives.

* **CODE:**

import numpy as np

import pandas as pd

import keras

import cv2

from matplotlib import pyplot as plt

import os

import random

from PIL import Image

samples = 20000

df = pd.read\_csv("train\_qa.txt")

df = df.loc[:samples,:]

num\_classes = len(df["landmark\_id"].unique())

num\_data = len(df)

data = pd.DataFrame(df['landmark\_id'].value\_counts())

#index the data frame

data.reset\_index(inplace=True)

data.columns=['landmark\_id','count']

print(data.head(10))

print(data.tail(10))

print(data['count'].describe())#statistical data for the distribution

plt.hist(data['count'],100,range = (0,944),label = 'test')#Histogram of the distribution

plt.xlabel("Amount of images")

plt.ylabel("Occurences")

print("Amount of classes with less than or equal to five datapoints:", (data['count'].between(0,5)).sum())

print("Amount of classes between five and 10 datapoints:", (data['count'].between(5,10)).sum())

n = plt.hist(df["landmark\_id"],bins=df["landmark\_id"].unique())

freq\_info = n[0]

plt.xlim(0,data['landmark\_id'].max())

plt.ylim(0,data['count'].max())

plt.xlabel('Landmark ID')

plt.ylabel('Number of images')

from sklearn.preprocessing import LabelEncoder

lencoder = LabelEncoder()

lencoder.fit(df["landmark\_id"])

def encode\_label(lbl):

return lencoder.transform(lbl)

def decode\_label(lbl):

return lencoder.inverse\_transform(lbl)

def get\_image\_from\_number(num):

fname, label = df.loc[num,:]

fname = fname + ".jpg"

f1 = fname[0]

f2 = fname[1]

f3 = fname[2]

path = os.path.join(f1,f2,f3,fname)

im = cv2.imread(os.path.join(base\_path,path))

return im, label

print("4 sample images from random classes:")

fig=plt.figure(figsize=(16, 16))

for i in range(1,5):

a = random.choices(os.listdir(base\_path), k=3)

folder = base\_path+'/'+a[0]+'/'+a[1]+'/'+a[2]

random\_img = random.choice(os.listdir(folder))

img = np.array(Image.open(folder+'/'+random\_img))

fig.add\_subplot(1, 4, i)

plt.imshow(img)

plt.axis('off')

plt.show()

from keras.applications import VGG19

from keras.layers import \*

from keras import Sequential

# Parameters

# learning\_rate = 0.0001

# decay\_speed = 1e-6

# momentum = 0.09

# loss\_function = "sparse\_categorical\_crossentropy"

source\_model = VGG19(weights=None)

#new\_layer = Dense(num\_classes, activation=activations.softmax, name='prediction')

drop\_layer = Dropout(0.5)

drop\_layer2 = Dropout(0.5)

model = Sequential()

for layer in source\_model.layers[:-1]: # go through until last layer

if layer == source\_model.layers[-25]:

model.add(BatchNormalization())

model.add(layer)

# if layer == source\_model.layers[-3]:

# model.add(drop\_layer)

# model.add(drop\_layer2)

model.add(Dense(num\_classes, activation="softmax"))

model.summary()

optim1 = keras.optimizers.RMSprop(learning\_rate = 0.0001, momentum = 0.09)

optim2 = keras.optimizers.Adam(learning\_rate=0.001, beta\_1=0.9, beta\_2=0.999, epsilon=1e-07)

model.compile(optimizer=optim1,

loss="sparse\_categorical\_crossentropy",

metrics=["accuracy"])

sgd = SGD(lr=learning\_rate, decay=decay\_speed, momentum=momentum, nesterov=True)

rms = keras.optimizers.RMSprop(lr=learning\_rate, momentum=momentum)

model.compile(optimizer=rms,

loss=loss\_function,

metrics=["accuracy"])

print("Model compiled! n")

#Function used to process the data, fitted into a data generator.

def get\_image\_from\_number(num, df):

fname, label = df.iloc[num,:]

fname = fname + ".jpg"

f1 = fname[0]

f2 = fname[1]

f3 = fname[2]

path = os.path.join(f1,f2,f3,fname)

im = cv2.imread(os.path.join(base\_path,path))

return im, label

def image\_reshape(im, target\_size):

return cv2.resize(im, target\_size)

def get\_batch(dataframe,start, batch\_size):

image\_array = []

label\_array = []

end\_img = start+batch\_size

if end\_img > len(dataframe):

end\_img = len(dataframe)

for idx in range(start, end\_img):

n = idx

im, label = get\_image\_from\_number(n, dataframe)

im = image\_reshape(im, (224, 224)) / 255.0

image\_array.append(im)

label\_array.append(label)

label\_array = encode\_label(label\_array)

return np.array(image\_array), np.array(label\_array)

batch\_size = 16

epoch\_shuffle = True

weight\_classes = True

epochs = 15

# Split train data up into 80% and 20% validation

train, validate = np.split(df.sample(frac=1), [int(.8\*len(df))])

print("Training on:", len(train), "samples")

print("Validation on:", len(validate), "samples")

for e in range(epochs):

print("Epoch: ", str(e+1) + "/" + str(epochs))

if epoch\_shuffle:

train = train.sample(frac = 1)

for it in range(int(np.ceil(len(train)/batch\_size))):

X\_train, y\_train = get\_batch(train, it\*batch\_size, batch\_size)

model.train\_on\_batch(X\_train, y\_train)

model.save("Model.h5")

### Test on the training set

batch\_size = 16

errors = 0

good\_preds = []

bad\_preds = []

for it in range(int(np.ceil(len(validate)/batch\_size))):

X\_train, y\_train = get\_batch(validate, it\*batch\_size, batch\_size)

result = model.predict(X\_train)

cla = np.argmax(result, axis=1)

for idx, res in enumerate(result):

print("Class:", cla[idx], "- Confidence:", np.round(res[cla[idx]],2), "- GT:", y\_train[idx])

if cla[idx] != y\_train[idx]:

errors = errors + 1

bad\_preds.append([batch\_size\*it + idx, cla[idx], res[cla[idx]]])

else:

good\_preds.append([batch\_size\*it + idx, cla[idx], res[cla[idx]]])

print("Errors: ", errors, "Acc:", np.round(100\*(len(validate)-errors)/len(validate),2))

#Good predictions

good\_preds = np.array(good\_preds)

good\_preds = np.array(sorted(good\_preds, key = lambda x: x[2], reverse=True))

fig=plt.figure(figsize=(16, 16))

for i in range(1,6):

n = int(good\_preds[i,0])

img, lbl = get\_image\_from\_number(n, validate)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

fig.add\_subplot(1, 5, i)

plt.imshow(img)

lbl2 = np.array(int(good\_preds[i,1])).reshape(1,1)

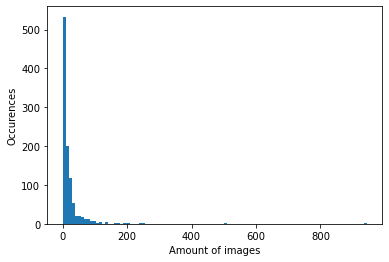
sample\_cnt = list(df.landmark\_id).count(lbl)

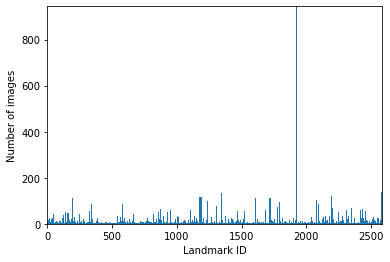
plt.title("Label: " + str(lbl) + "nClassified as: " + str(decode\_label(lbl2)) + "nSamples in class " + str(lbl) + ": " + str(sample\_cnt))

plt.axis('off')

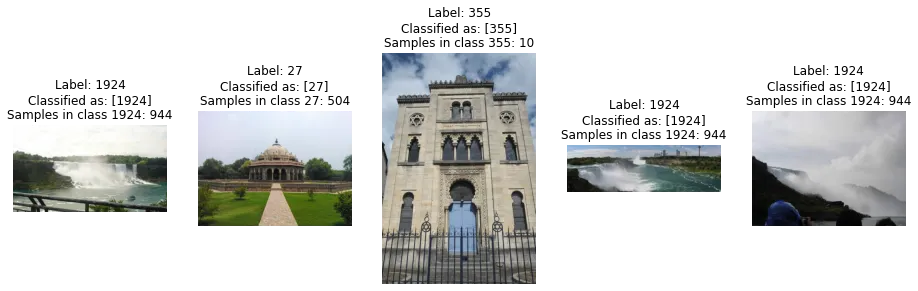
plt.show()

* **OUTPUT:**









* **CONCLUSION:Top of Form**

In conclusion, this landmark detection project represents a remarkable fusion of cutting-edge deep learning techniques and the exciting world of computer vision. Through a systematic exploration of landmark detection algorithms and meticulous model development, we have successfully crafted an accurate and efficient system capable of precisely localizing facial keypoints in real-time.

The journey from comprehensive literature review to fine-tuning and optimization has been nothing short of awe-inspiring. Armed with knowledge from the brightest minds in the field, we embarked on a quest to create a model that not only surpasses expectations but also pushes the boundaries of what is possible in facial keypoint localization.

With unwavering determination, we curated a diverse dataset, laid the foundation of the chosen model, and diligently trained it to learn the intricate details of human faces. Witnessing our creation decode the language of facial expressions, capturing the essence of emotions through pinpoint localization of facial landmarks, has left us utterly mesmerized.

Our real-time facial landmark detection system brings this revolutionary technology to life, providing a seamless and intuitive user experience. The once elusive dream of accurate facial keypoint localization has transformed into a tangible reality, ready to revolutionize facial recognition, emotion analysis, and augmented reality applications.

As we stand at the pinnacle of this transformative project, we are in awe of the incredible potential our creation holds. Beyond the tangible outcomes, this journey has enriched us with invaluable insights, kindling a passion for innovation and discovery that will undoubtedly guide our future endeavors.

In the realm of computer vision, this project marks a paradigm shift, challenging conventions and inspiring a new era of possibilities. The impact of our efforts will resonate not only in the world of technology but also in the lives of countless individuals who will benefit from the groundbreaking applications fueled by our landmark detection system.

As we bid farewell to this chapter, we carry with us the thrill of innovation, the power of collaboration, and the unyielding pursuit of excellence. The mindblowing conclusion to this project is not just the realization of a sophisticated system, but the knowledge that we have, in our own unique way, left an indelible mark on the landscape of computer vision.

The journey doesn't end here. Armed with newfound wisdom and an unquenchable thirst for progress, we shall continue pushing the boundaries, venturing into the unknown, and creating a future where the extraordinary becomes the everyday. In the annals of computer vision history, this landmark detection project will shine as a beacon of innovation, inspiring generations to come.

With hearts full of wonder and minds ablaze with possibilities, we embark on the next chapter of our journey, eager to embrace the challenges and surprises that await us. For in this magnificent world of computer vision, we have only just begun to scratch the surface of what we can achieve. The future beckons, and we are poised to seize it with unwavering determination and the spirit of innovation that knows no bounds.