Cine-Collab

Connect | Create | Collabarate

Optimized Skill Endorsement Web based solution for the Entertainment Industry.

SLIIT FACULTY OF COMPUTING

3/4/2024

Team

Supervisor :
Co-Supervisor:

Perera K.C.M

Abeyrathna H.K.H

Raja.R.K.K

M.A.R.F.M FAZIL

Introduction

- What is Cine-Collab?
- How to find Opportunities ?
- What are the problems of existing platforms ?
- How to give most accurate output?

Overall Project Description

- ❖ importance of collaboration in the entertainment industry
- ❖ growing interest in the local cinema industry, opportunities for them are limited
- ❖A more collaborative and inclusive approach is needed to foster growth
- ❖provision of a suitable platform for deserving talents

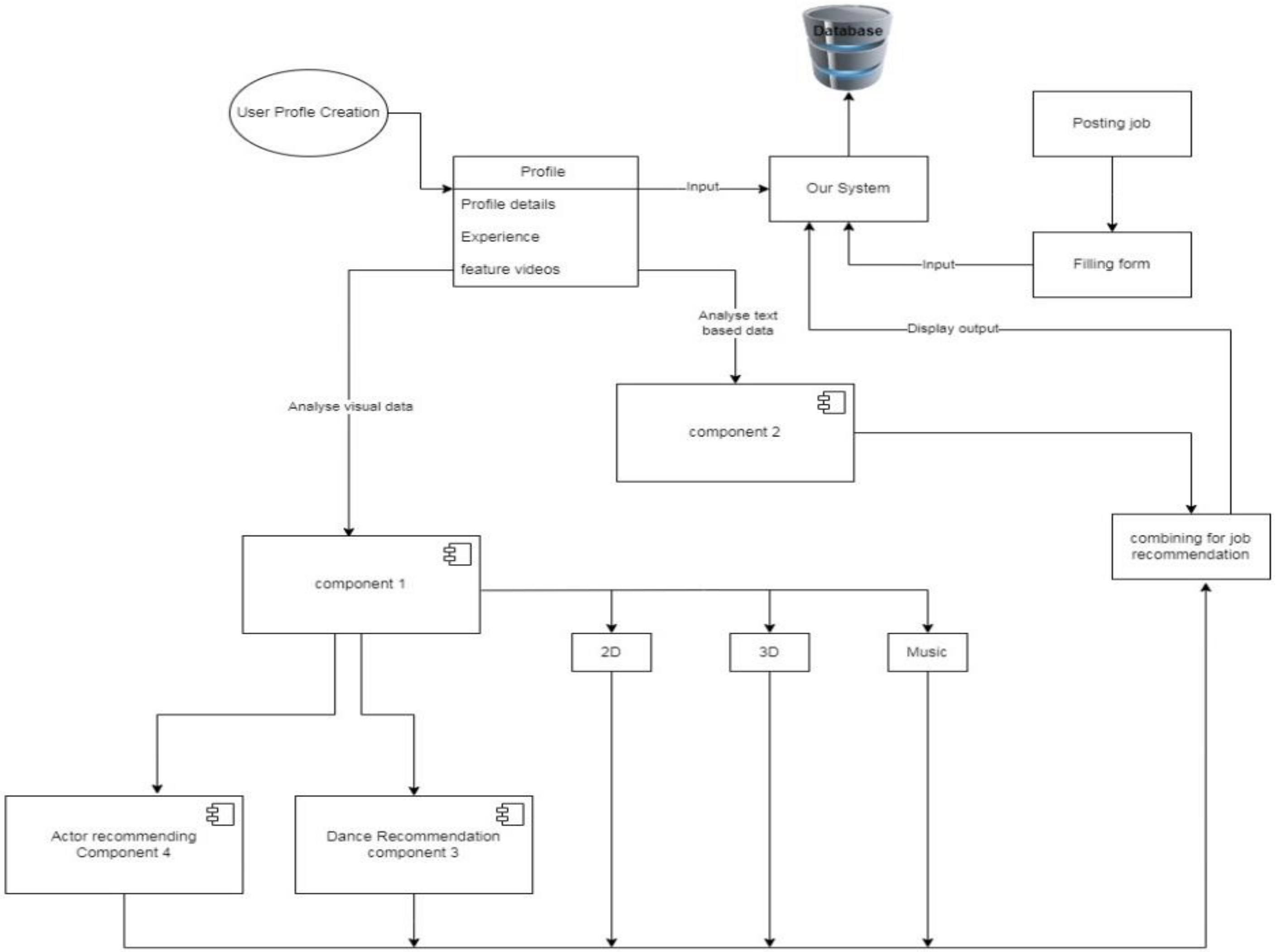
Objective

Main Objective: Solving issues related to showcasing skills and finding opportunities and collaborating issues in Entertainment Industry

Sub Objectives:

- Sub Objective 1:
- Sub Objective 2:
- Sub Objective 3: Facilitate the identification and selection of skilled dancers for specific dance-related job roles in the entertainment industry.
- Sub Objective 4: Simplifying character role auditions and providing more accurate job recommendations by analyzing visual data.

System Overall Diagram



Research and Background Guidance

Gorden De Silva
Director & Cinematographer
(Rupawahini)

Abishek Palraj
Film maker&
 Cinematographer

(Independent)

Gorden De Silva
Actor & TV host
(Maharaja Network)

IT20638504 | Perera K.C.M

Video classification & job recommendation

Bachelor of Science (Hons) in Information Technology Specializing in Interactive Media

Research Problem

- Limited industry exposure .
- Lack of accessible platforms.
- Difficulty in Finding Opportunities according to their specific talents.
- Unidentified talent potential.

Research Questions

- How to classify job roles ?

- How to suggest relevant job matches to the talents?

- How to recognize specific talents according to job rolls?

Research Gap

Classifying roles for Acting and Job recommendation

- ❖ Existing Platforms Do not Has a Feature to categorize uploaded videos/pictures according to type (live action , VFX , 2D/3D animation)

- ❖ integration of video and image analysis techniques, enabling the platform to understand visual content.

- ❖ Focus facial regions and Improving Face recognition on Shaky frames

Specific Objective

Video classification and job recommendation system

- ❖ To classifying similarities, types (genres) between uploaded videos/images by the user and recommend the best match.

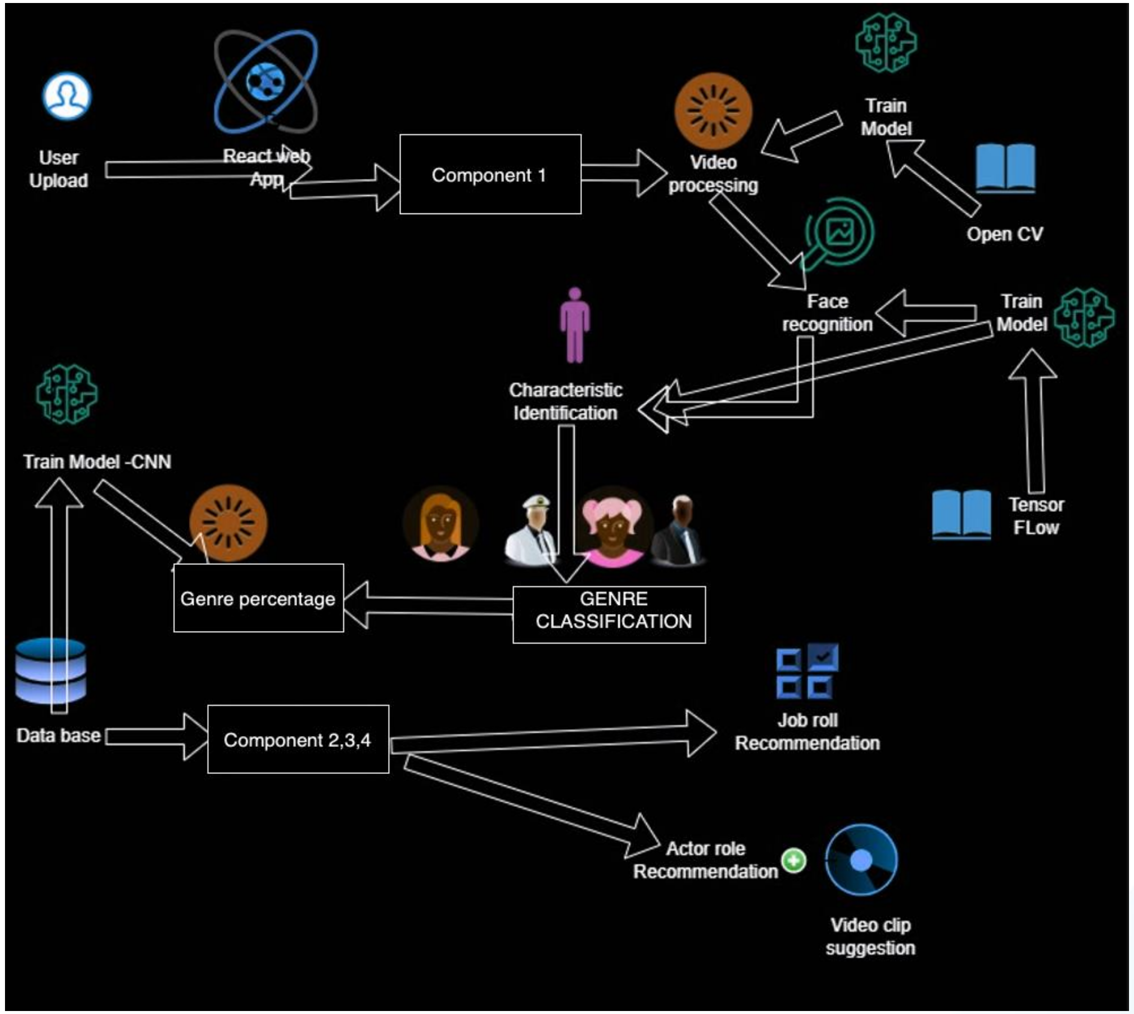
Sub Objectives

Video classification and job recommendation system

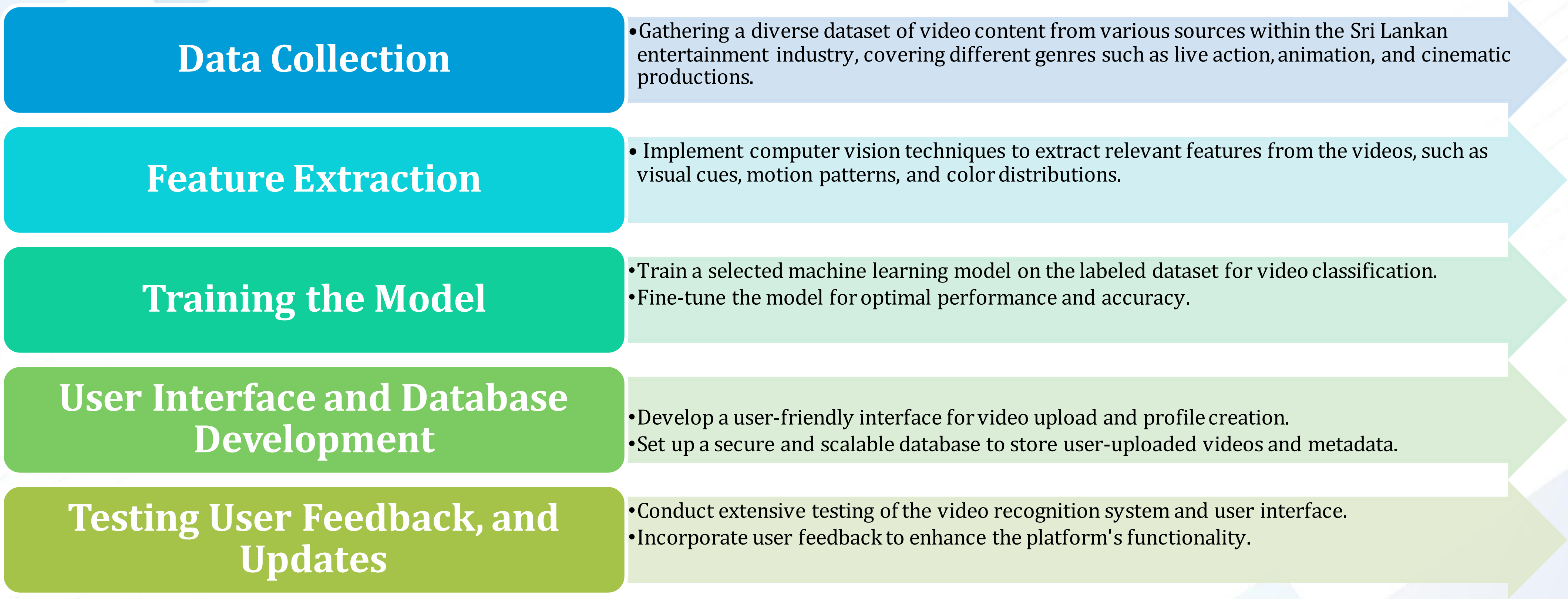
- ❖ Develop a user-friendly interface for talents to upload their creations as videos or images.
- ❖ Implement video and image analysing techniques to process and categorize the uploaded content.
- ❖ Optimize the recommendation system to handle diverse talents effectively, ensuring that talents with unique artistic skills are appropriately matched with suitable job opportunities.

IT20638504 | Perera K.C.M| TMP-2023-24-109

System overview diagram



Methodology



Data Collection

- Gathering a diverse dataset of video content from various sources within the Sri Lankan entertainment industry, covering different genres such as live action, animation, and cinematic productions.

Feature Extraction

- Implement computer vision techniques to extract relevant features from the videos, such as visual cues, motion patterns, and color distributions.

Training the Model

- Train a selected machine learning model on the labeled dataset for video classification.
- Fine-tune the model for optimal performance and accuracy.

User Interface and Database Development

- Develop a user-friendly interface for video upload and profile creation.
- Set up a secure and scalable database to store user-uploaded videos and metadata.

Testing User Feedback, and Updates

- Conduct extensive testing of the video recognition system and user interface.
- Incorporate user feedback to enhance the platform's functionality.

Research Gap Summary

Feature	A	B	C	D	Proposed S ystem
video classification with motion Detection.	✗	✓	✓	✗	✓
Motion Compensation	✓	✗	✓	✓	✓
Object Detection	✓	✓	✗	✗	✓
Image recognition & video processing to classify genres	✗	✓	✗	✓	✓

Tools and Technologies

Technologies:

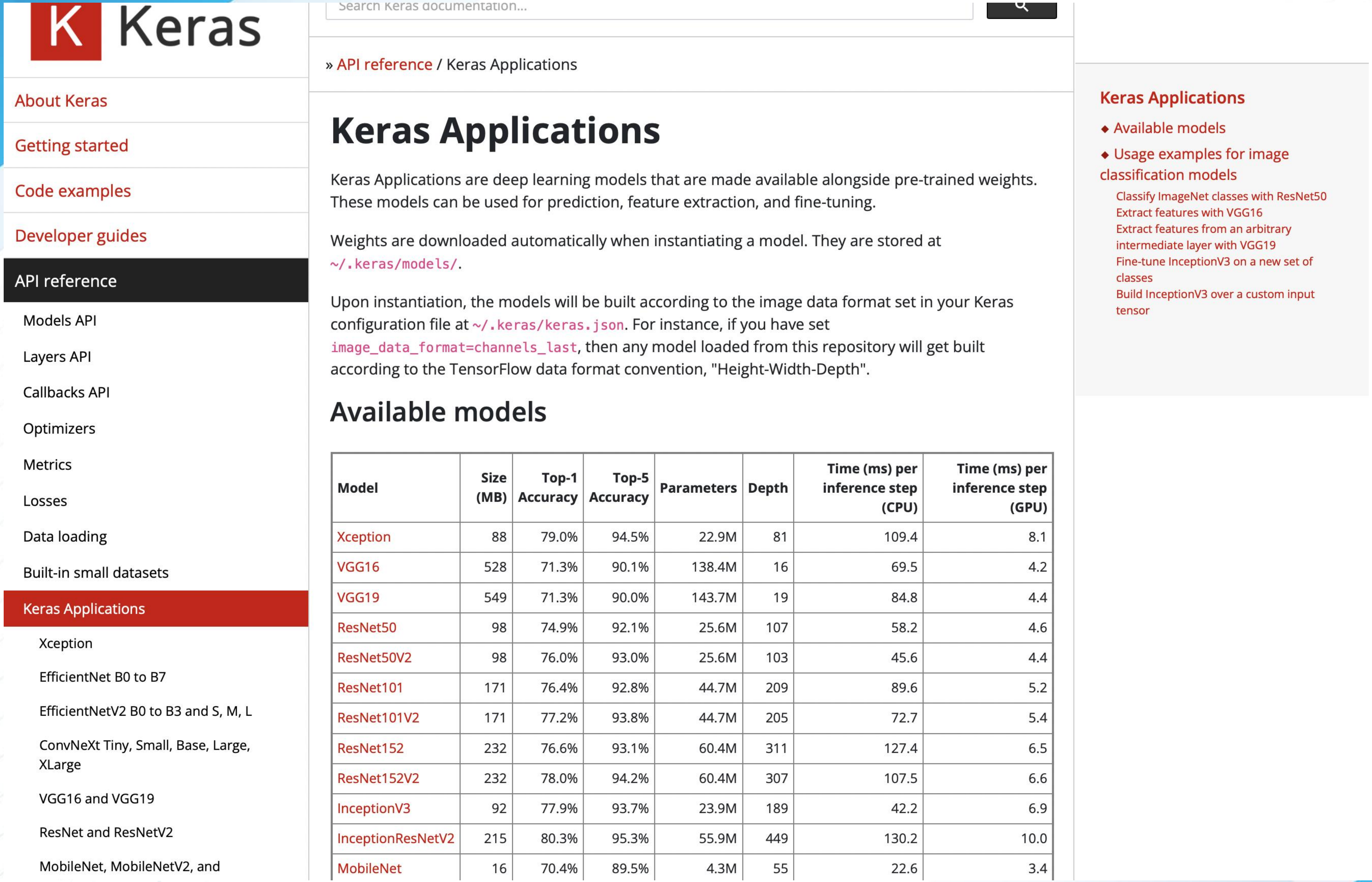
- **CNN – VIDEO FRAME PROCESSING**
- **Inception V3**

Tools:

- ❖ Machine learning – Python
- ❖ Version Controlling – Git Hub
- ❖ For implementation needs Google collab
- ❖ Front end – React JS
- ❖ Database - mongo DB

Specific Objective:

- ❖ keras
- ❖ Video processing – Open CV


Keras
Search Keras documentation...
About Keras
Getting started
Code examples
Developer guides
API reference
Models API
Layers API
Callbacks API
Optimizers
Metrics
Losses
Data loading
Built-in small datasets
Keras Applications
Xception
EfficientNet B0 to B7
EfficientNetV2 B0 to B3 and S, M, L
ConvNeXt Tiny, Small, Base, Large, XLarge
VGG16 and VGG19
ResNet and ResNetV2
MobileNet, MobileNetV2, and
» API reference / Keras Applications
Keras Applications
◆ Available models
◆ Usage examples for image classification models
Classify ImageNet classes with ResNet50
Extract features with VGG16
Extract features from an arbitrary intermediate layer with VGG19
Fine-tune InceptionV3 on a new set of classes
Build InceptionV3 over a custom input tensor

Keras Applications

Keras Applications are deep learning models that are made available alongside pre-trained weights. These models can be used for prediction, feature extraction, and fine-tuning.

Weights are downloaded automatically when instantiating a model. They are stored at `~/.keras/models/`.

Upon instantiation, the models will be built according to the image data format set in your Keras configuration file at `~/.keras/keras.json`. For instance, if you have set `image_data_format=channels_last`, then any model loaded from this repository will get built according to the TensorFlow data format convention, "Height-Width-Depth".

Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4

References

A.[**Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks"** by Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. (IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017)

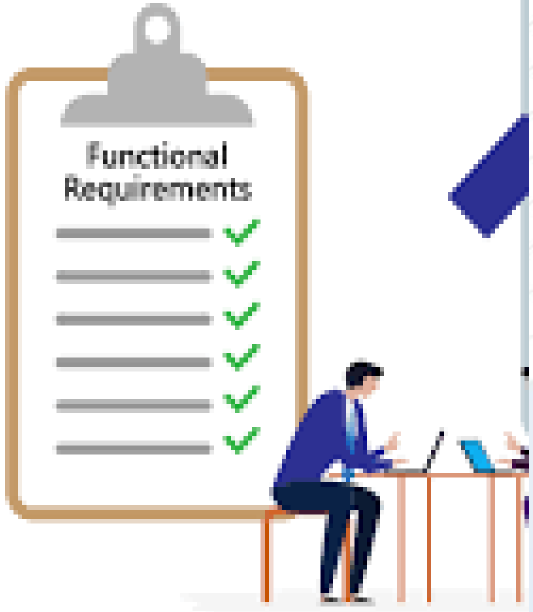
B."You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. (Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016)

C."Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. (Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016)

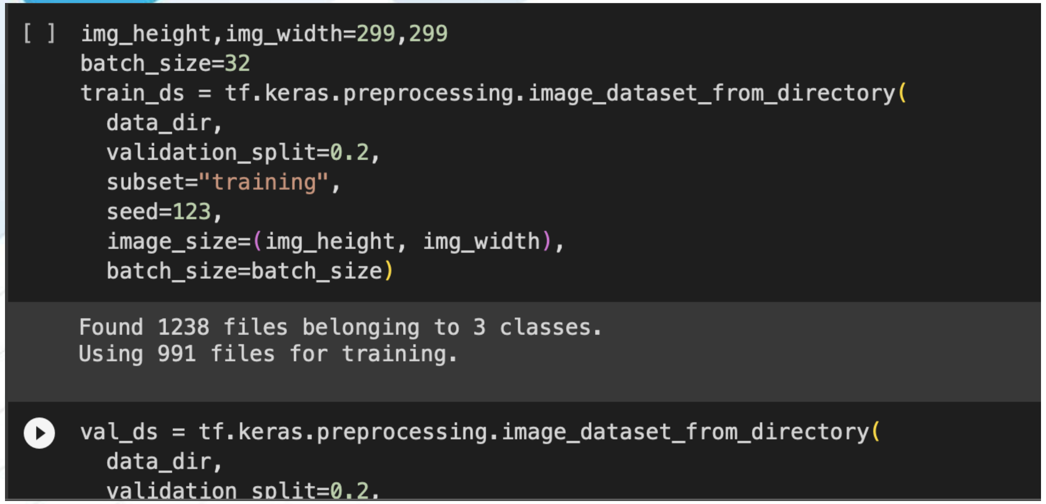
D. "ImageNet Classification with Deep Convolutional Neural Networks" by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. (Advances in Neural Information Processing Systems, 2012)

Functional Requirements

- ✓ **Accurate Video Categorization**
- ✓ **Efficient Model Training**
- ✓ **User-Friendly Upload**
- ✓ **Accurate job recommendations**

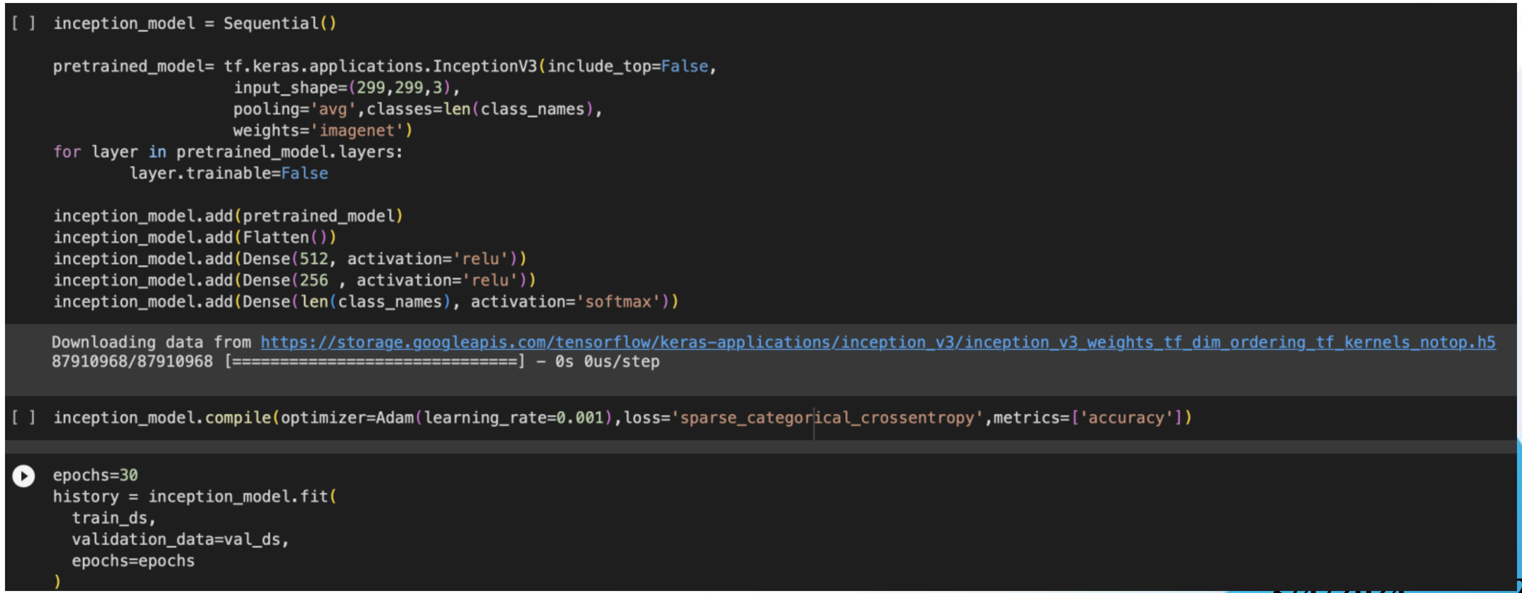


DATA COLLECTION AND PRE PROCESSING

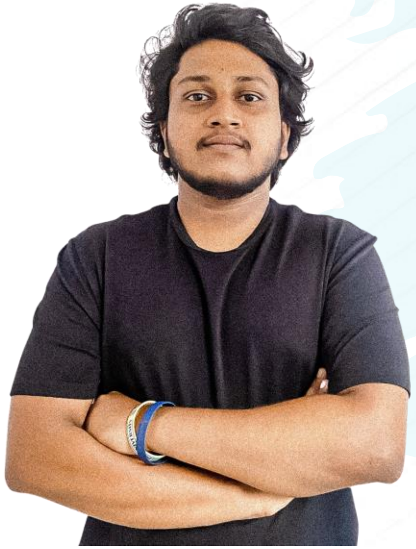


```
[ ]  img_height,img_width=299,299
     batch_size=32
     train_ds = tf.keras.preprocessing.image_dataset_from_directory(
       data_dir,
       validation_split=0.2,
       subset="training",
       seed=123,
       image_size=(img_height, img_width),
       batch_size=batch_size)

     Found 1238 files belonging to 3 classes.
     Using 991 files for training.

     val_ds = tf.keras.preprocessing.image_dataset_from_directory(
       data_dir,
       validation_split=0.2,
```

MODEL TRAINING



```
[ ]  inception_model = Sequential()

     pretrained_model= tf.keras.applications.InceptionV3(include_top=False,
                        input_shape=(299,299,3),
                        pooling='avg',classes=len(class_names),
                        weights='imagenet')
     for layer in pretrained_model.layers:
             layer.trainable=False

     inception_model.add(pretrained_model)
     inception_model.add(Flatten())
     inception_model.add(Dense(512, activation='relu'))
     inception_model.add(Dense(256 , activation='relu'))
     inception_model.add(Dense(len(class_names), activation='softmax'))

     Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
     87910968/87910968 [==============================] - 0s 0us/step

[ ]  inception_model.compile(optimizer=Adam(learning_rate=0.001),loss='sparse_categorical_crossentropy',metrics=['accuracy'])

     epochs=30
     history = inception_model.fit(
       train_ds,
       validation_data=val_ds,
       epochs=epochs
     )
```



IT20166274 | Abeyrathna H.K.H

Job Posting and Skill Matching

Bachelor of Science (Hons) in Information Technology Specializing in Interactive Media

Research problem

How can a skill matching algorithm be used to compare job requirements with user profiles and identify potential matches?

Objectives

- Specific Objective:
- Develop a job posting and skill matching component that utilizes machine learning algorithms to extract and analyze textual data from user profiles, information sections, and post captions to suggest relevant profiles for job postings within a web application.

- Main Objective:
- The main objective of this component is to enhance user experience and foster skill matching within the web application by:
- 1. Extracting and analyzing textual data from user profiles and posts.
- 2. Utilizing machine learning techniques for keyword extraction and data analysis.
- 3. Providing job posters with suggestions of relevant profiles based on their job requirements.

Technologies to be used..

- Web - react
- Main Back end: Node Express framework
- ML backend : FastAPI framework
- Database: MongoDB
- Scikit learn

Functional and non - functional requirements

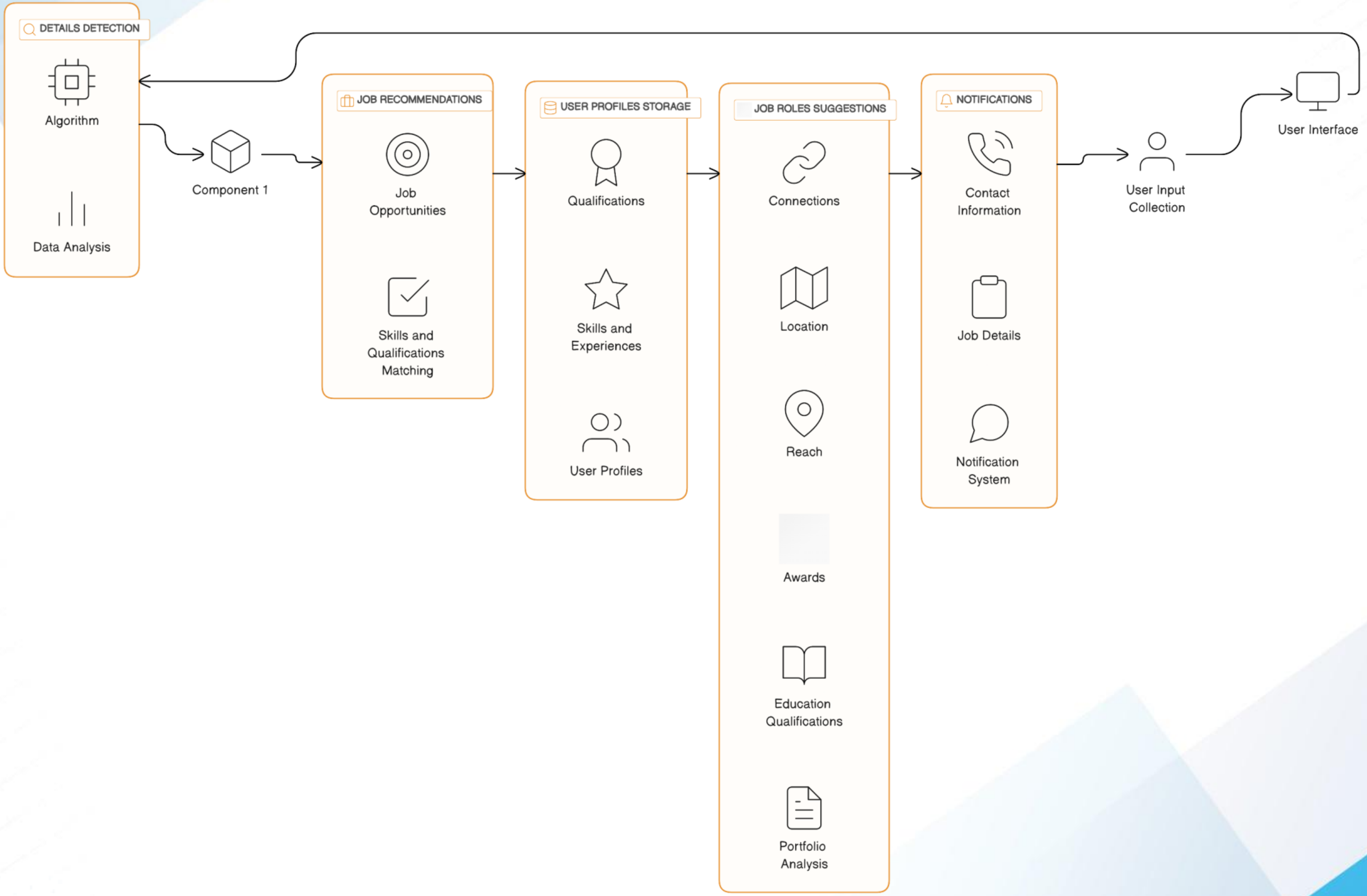
Functional Requirements:

- User Registration and Profile Creation
- User Input Collection
- Data Analysis
- Job Recommendations
- Data Storage and Organization

Non-Functional Requirements:

- User-Friendly Interface
- Performance
- Security
- Reliability
- Accessibility
- Data Privacy

System overview diagram

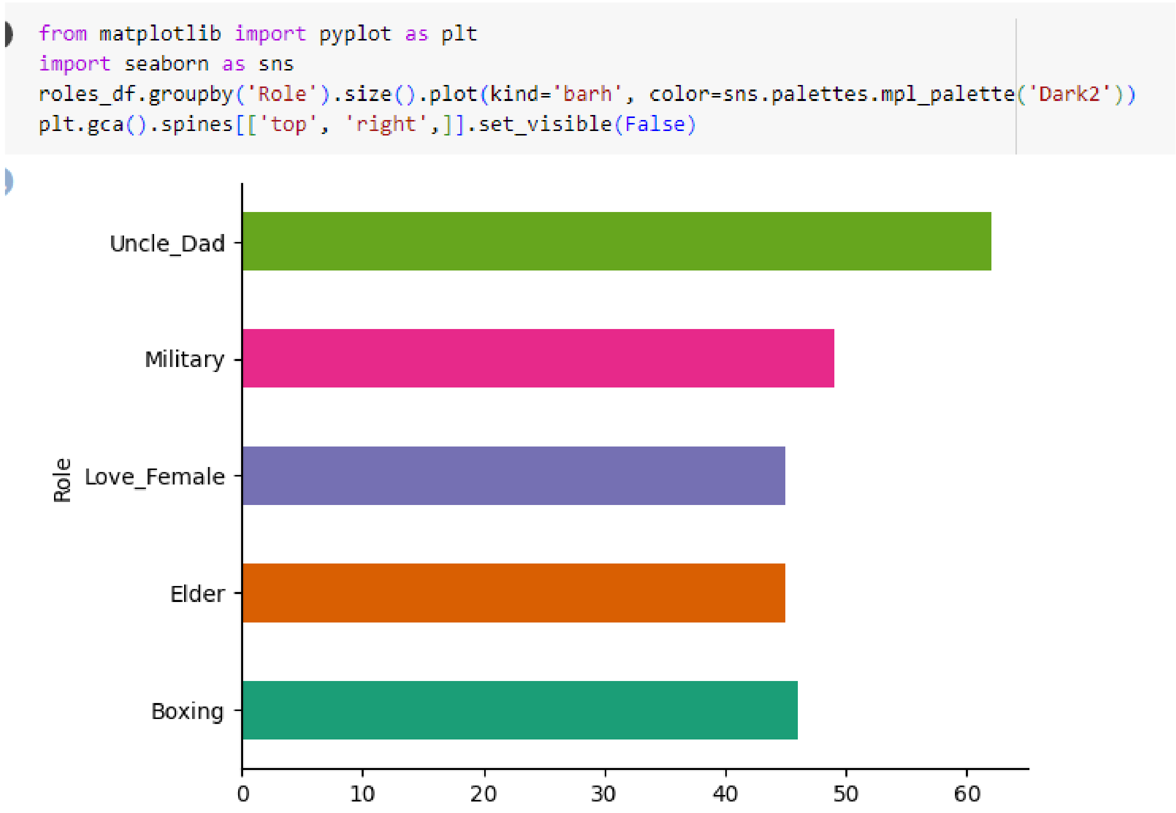


Research gap

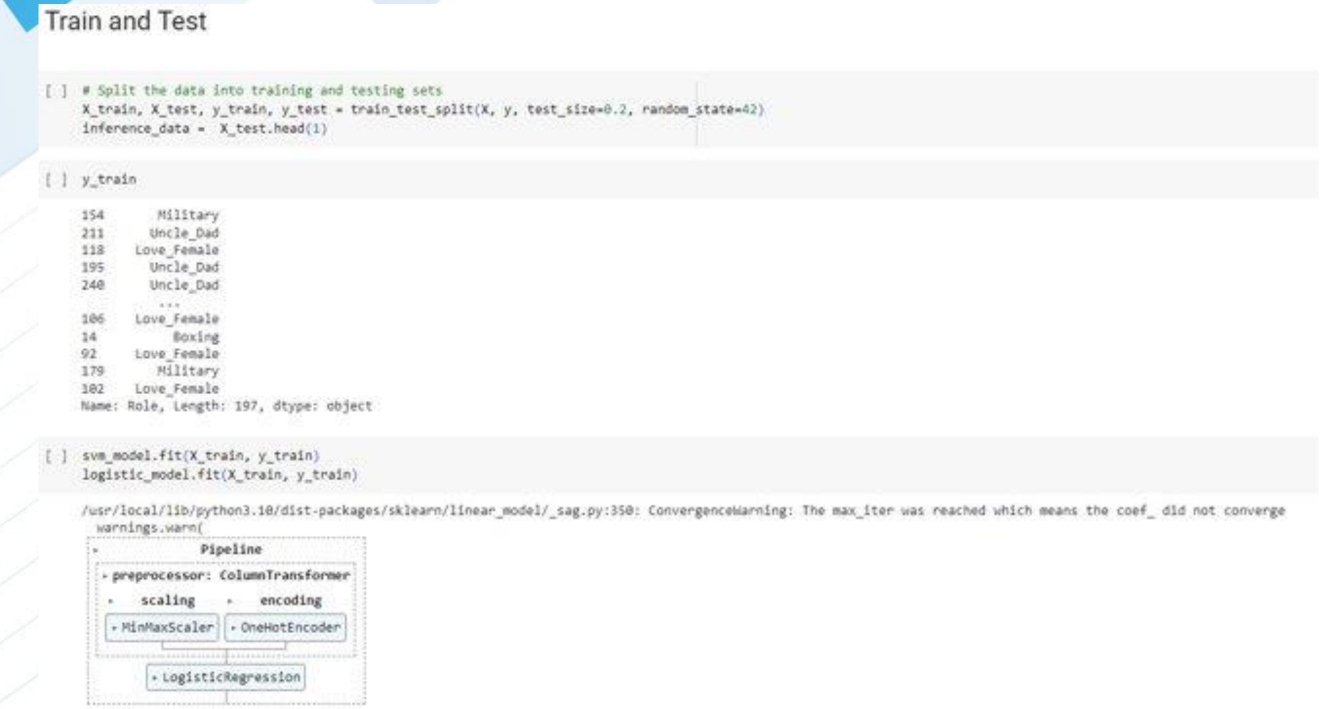
- **Focus on Entertainment Industry**: The research gap lies in the lack of dedicated platforms that focus on the unique challenges and opportunities in the entertainment industry.

- **Limited Utilization of Advanced Technologies**: There is a research gap in not fully leveraging and data analysis techniques to enhance job recommendations and skill matching in the entertainment industry.

Evidence



```
from matplotlib import pyplot as plt
import seaborn as sns
roles_df.groupby('Role').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```



Train and Test

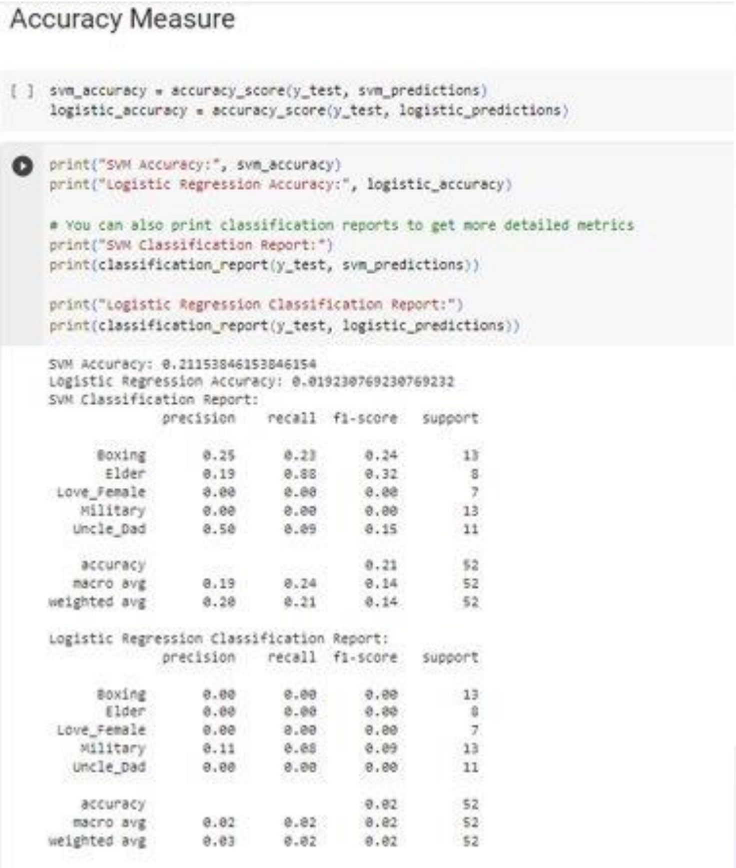
```
[ ] # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    inference_data =  X_test.head(1)
```

```
[ ] y_train
```

```
154        Military
211      Uncle_Dad
118    Love_Female
195      Uncle_Dad
240      Uncle_Dad
          ...
106    Love_Female
14           Boxing
92     Love_Female
179        Military
102    Love_Female
Name: Role, Length: 197, dtype: object
```

```
[ ] svm_model.fit(X_train, y_train)
    logistic_model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn(
```

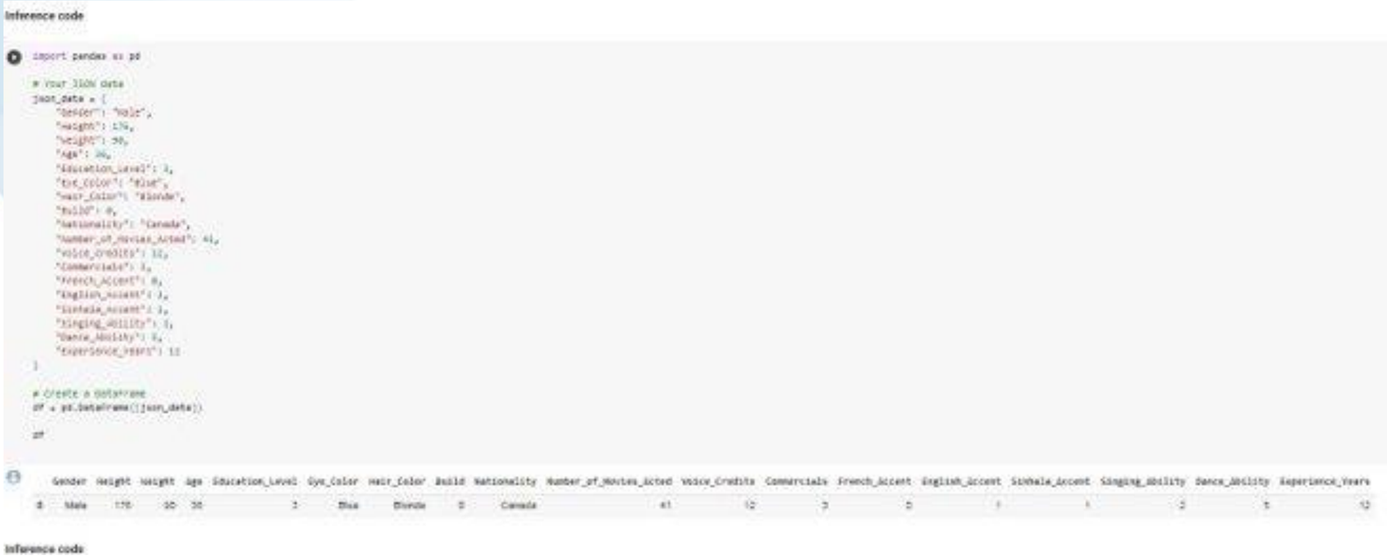


Accuracy Measure

```
[ ] svm_accuracy = accuracy_score(y_test, svm_predictions)
    logistic_accuracy = accuracy_score(y_test, logistic_predictions)
```

```
print("SVM Accuracy:", svm_accuracy)
print("Logistic Regression Accuracy:", logistic_accuracy)

# You can also print classification reports to get more detailed metrics
print("SVM Classification Report:")
print(classification_report(y_test, svm_predictions))

print("Logistic Regression Classification Report:")
print(classification_report(y_test, logistic_predictions))
```

```
SVM Accuracy: 0.21153846153846154
Logistic Regression Accuracy: 0.019230769230769232
SVM Classification Report:
              precision    recall  f1-score   support

      Boxing       0.25      0.23      0.24        13
       Elder       0.19      0.88      0.32         8
 Love_Female       0.00      0.00      0.00         7
    Military       0.00      0.00      0.00        13
   Uncle_Dad       0.50      0.09      0.15        11

    accuracy                           0.21        52
   macro avg       0.19      0.24      0.14        52
weighted avg       0.20      0.21      0.14        52

Logistic Regression Classification Report:
              precision    recall  f1-score   support

      Boxing       0.00      0.00      0.00        13
       Elder       0.00      0.00      0.00         8
 Love_Female       0.00      0.00      0.00         7
    Military       0.11      0.08      0.09        13
   Uncle_Dad       0.00      0.00      0.00        11

    accuracy                           0.02        52
   macro avg       0.02      0.02      0.02        52
weighted avg       0.03      0.02      0.02        52
```

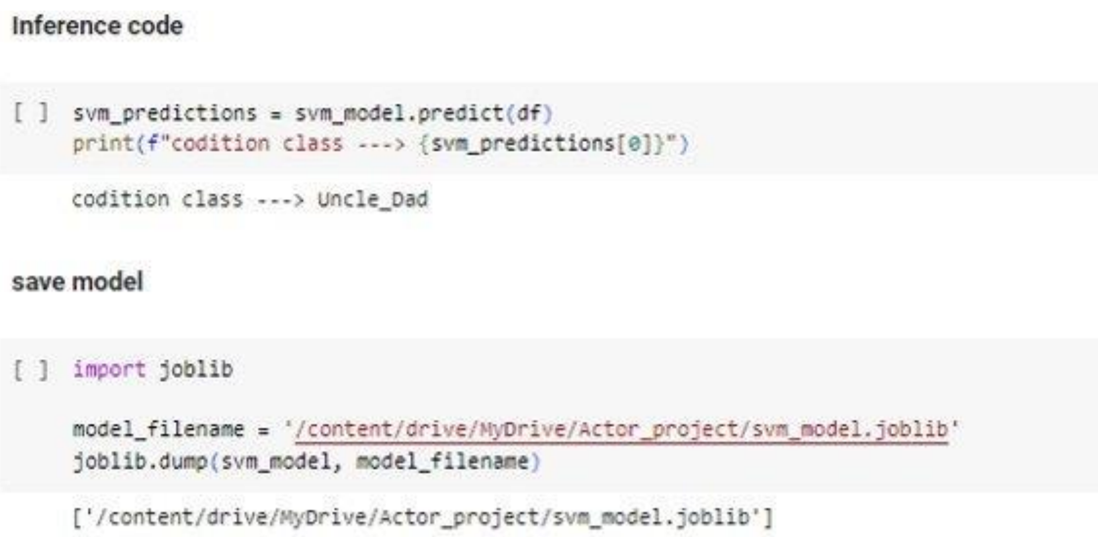


Inference code

```
import pandas as pd

# Your JSON data
json_data = {
    "Gender": "Male",
    "Height": 170,
    "Weight": 50,
    "Age": 30,
    "Education_Level": 3,
    "Eye_Color": "Blue",
    "Hair_Color": "Blonde",
    "Build": 0,
    "Nationality": "Canada",
    "Number_of_Movies_Acted": 41,
    "Voice_Credits": 12,
    "Commercials": 3,
    "French_Accent": 0,
    "English_Accent": 1,
    "Sinhala_Accent": 1,
    "Singing_Ability": 2,
    "Dance_Ability": 5,
    "Experience_Years": 12
}

# Create a DataFrame
df = pd.DataFrame([json_data])

df
```

```
  Gender Height Weight Age Education_Level Eye_Color Hair_Color Build Nationality Number_of_Movies_Acted Voice_Credits Commercials French_Accent English_Accent Sinhala_Accent Singing_Ability Dance_Ability Experience_Years
0   Male    170     50  30               3      Blue     Blonde     0      Canada                     41            12           3             0              1              1               2             5               12
```

Inference code



Inference code

```
[ ] svm_predictions = svm_model.predict(df)
    print(f"codition class ---> {svm_predictions[0]}")
```

```
codition class ---> Uncle_Dad
```

save model

```
[ ] import joblib

    model_filename = '/content/drive/MyDrive/Actor_project/svm_model.joblib'
    joblib.dump(svm_model, model_filename)
```

```
['/content/drive/MyDrive/Actor_project/svm_model.joblib']
```

Research Gap Summary

Feature	1	2	3		Proposed System
Improve accuracy on domain entity extraction.	✗	✓	✗		✓
Detect accuracy of user input details.	✓	✓	✓		✓
Skill Matching with Specific Requirements	✗	✗	✗		✓
Multi-Dimensional Talent Evaluation	✓	✗	✗		✓

Methodology

DATA COLLECTION

DATA ANALYSIS

ENHANCE ALGORITHMS

DATA INTEGRATION

EVALUATION

References

[1].A. -N. Lee, K. -Y. Chen and C. -T. Li, "ActRec: A Word Embedding-based Approach to Recommend Movie Actors to Match Role Descriptions," 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), The Hague, Netherlands, 2020, pp. 389-392, doi: 10.1109/ASONAM49781.2020.9381452.

[2]K. Appadoo, M. B. Soonnoo and Z. Mungloo-Dilmohamud, "Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing," 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 2020, pp. 1-6, doi: 10.1109/CSDE50874.2020.9411584.

[3]Peng Yi, C. Yang, Chen Li and Y. Zhang, "A job recommendation method optimized by position descriptions and resume information," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 2016, pp. 761-764, doi: 10.1109/IMCEC.2016.7867312.

IT20667146 | RAJA.R.K.K

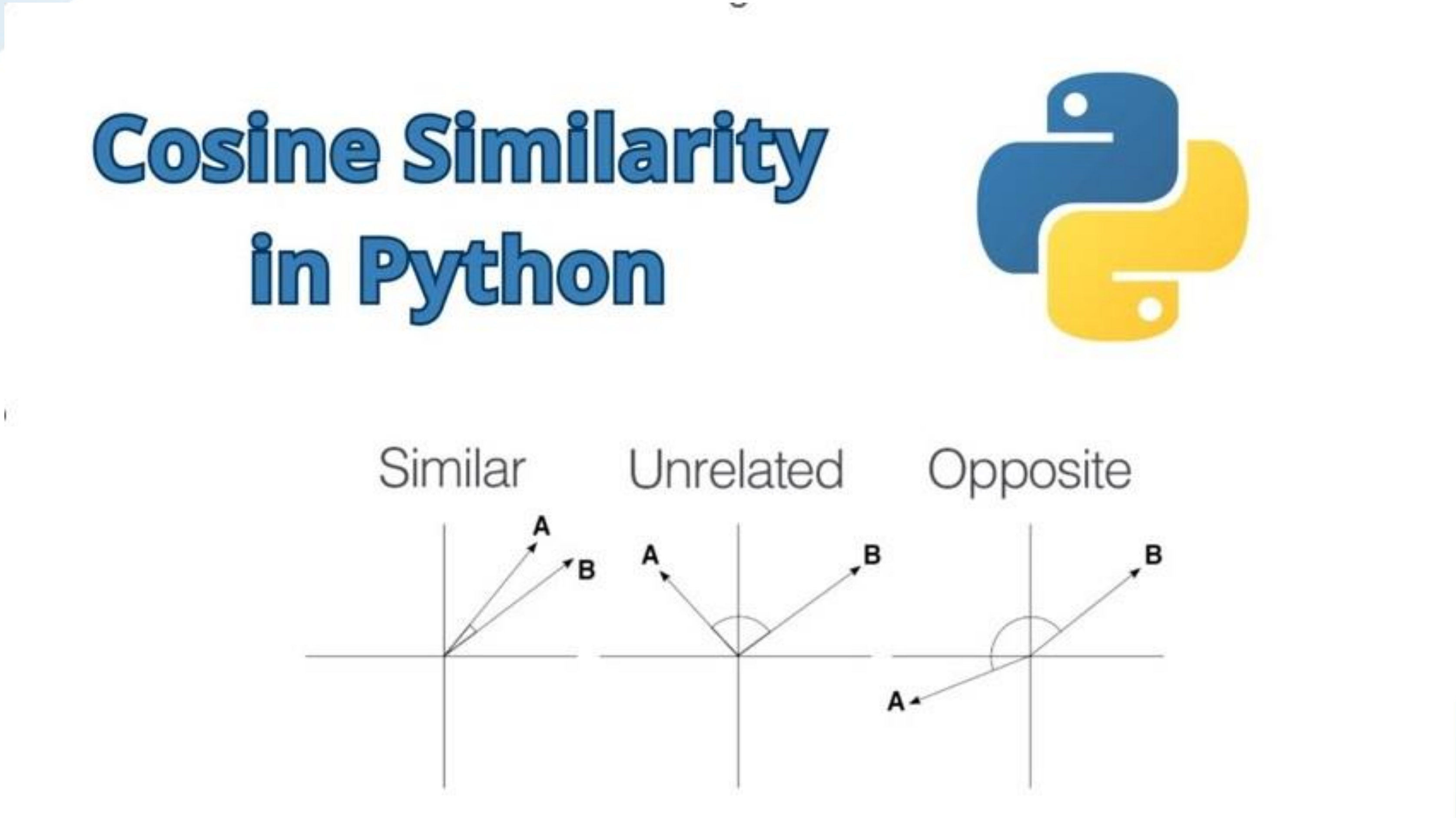
Dance Demo Video Upload, Filtering and Recommending .

Bachelor of Science (Hons) in Information Technology Specializing in Interactive Media

Specific and Sub Objectives

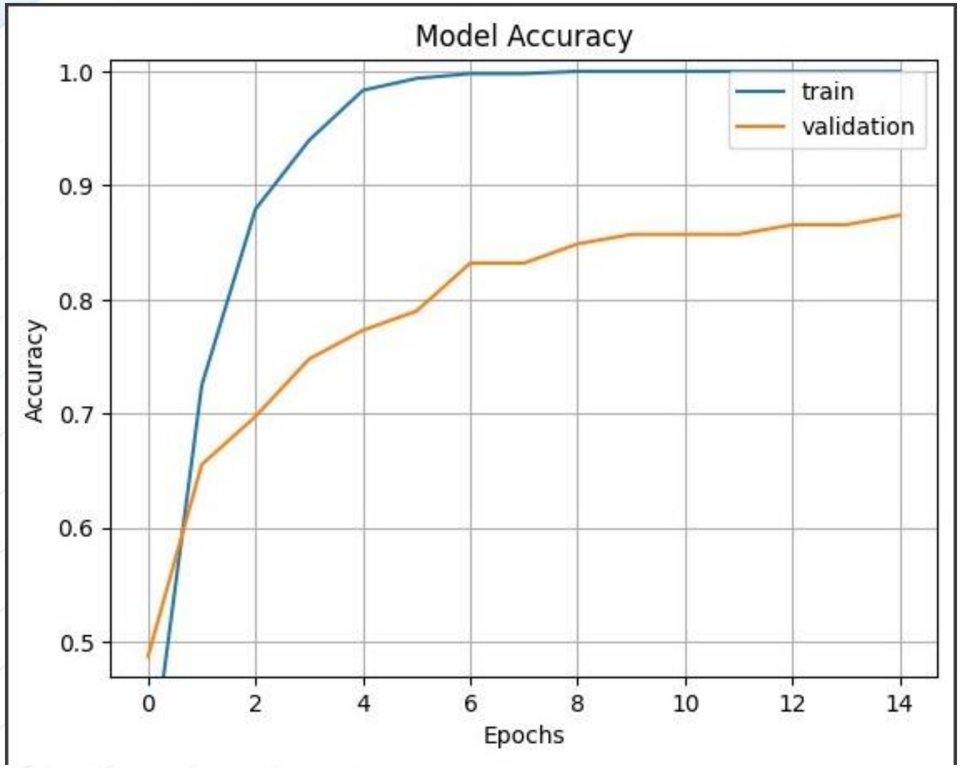
Dance Demo Video Upload , Filtering , and Recommending

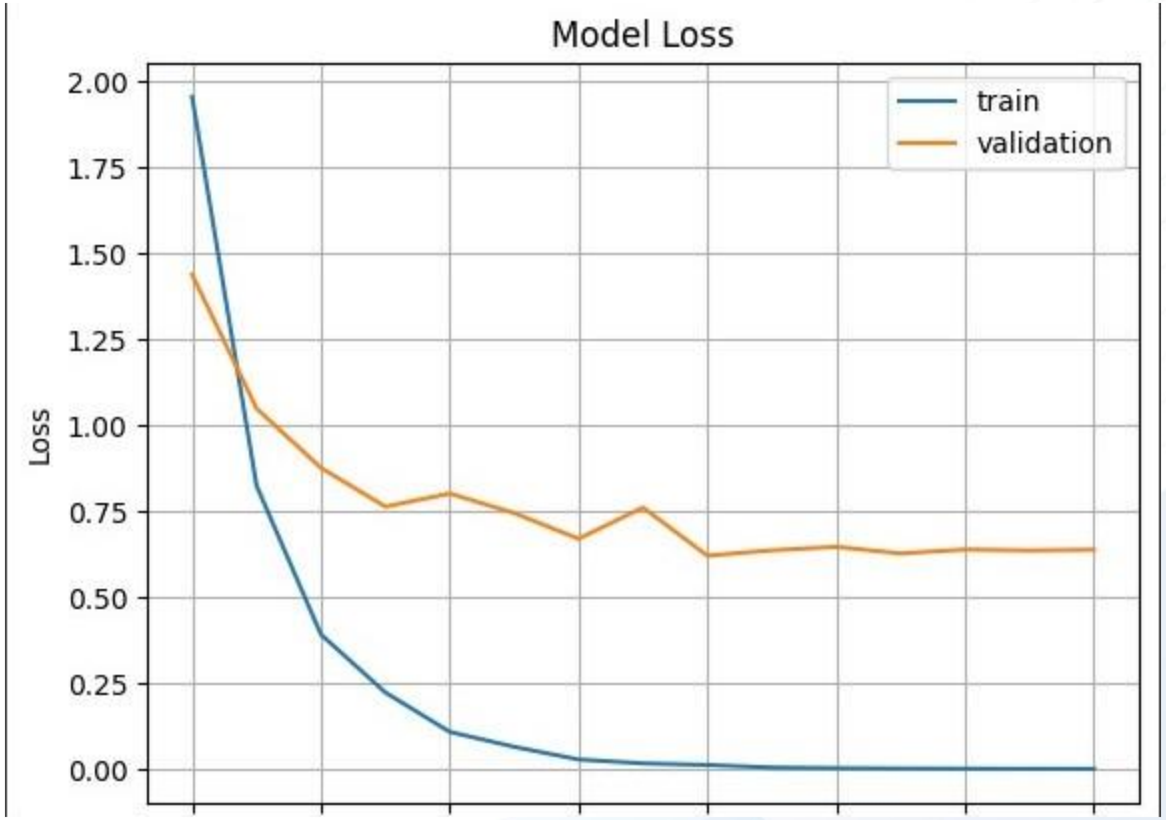
- ❖ Track the similarity between the original demo video and the dance videos submitted by job applicants.
- ❖ Utilization of video analysis techniques and machine learning algorithms to compare.
- ❖ Dance movements, gestures, and styles with consideration for Sri Lankan and South Indian cultural elements.
- ❖ The uploaded video will recommend to the similar role hiring clients or companys.

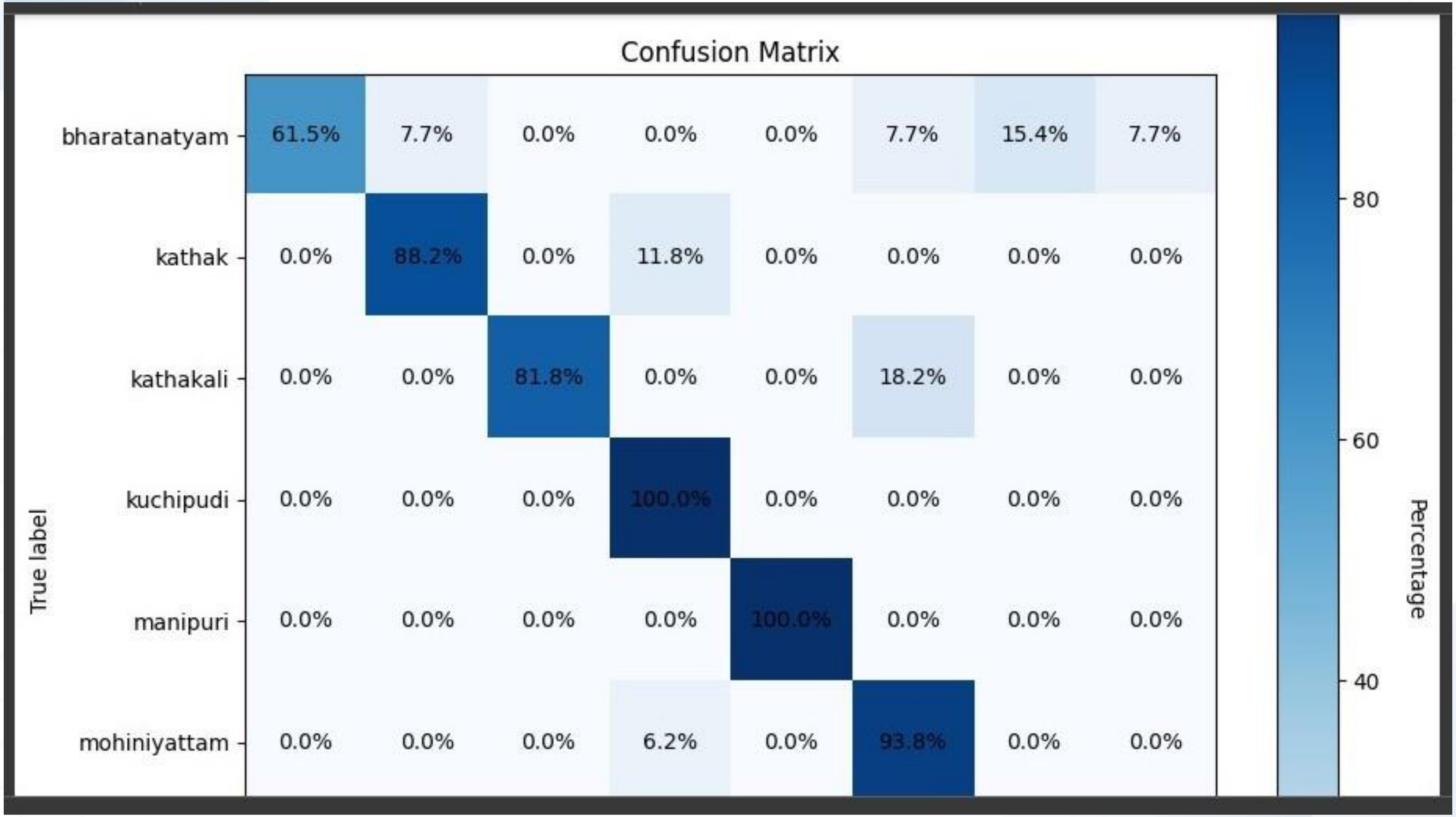


Cosine Similarity in Python

CNN ARCHITECTURES EVIDENCE OF COMPLETION



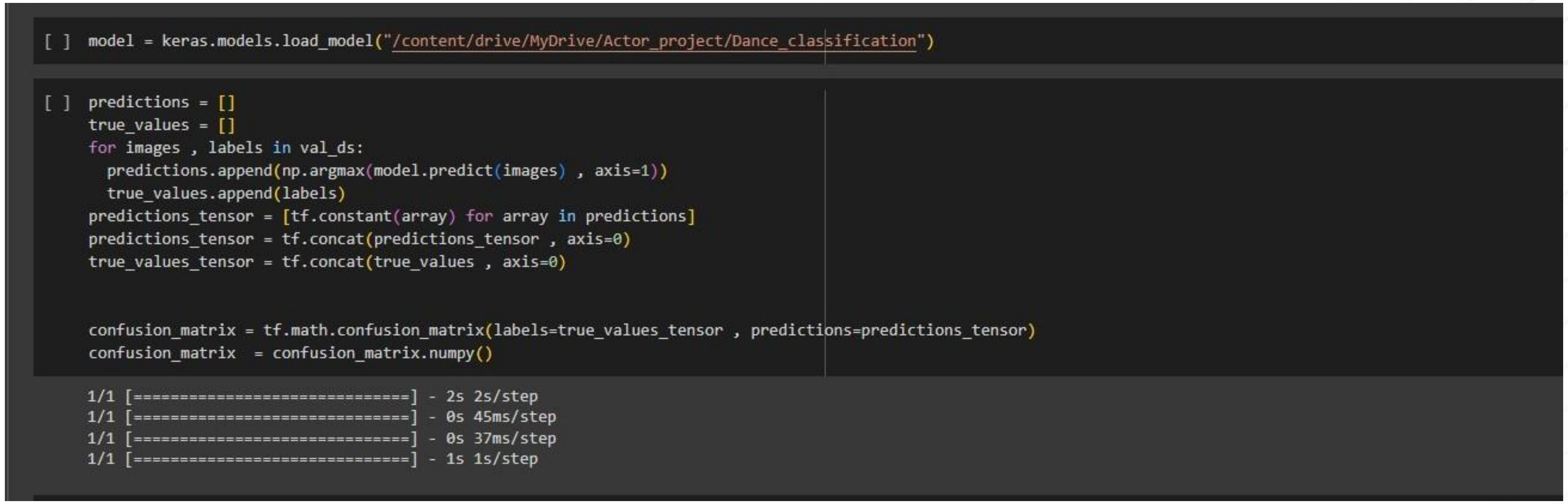




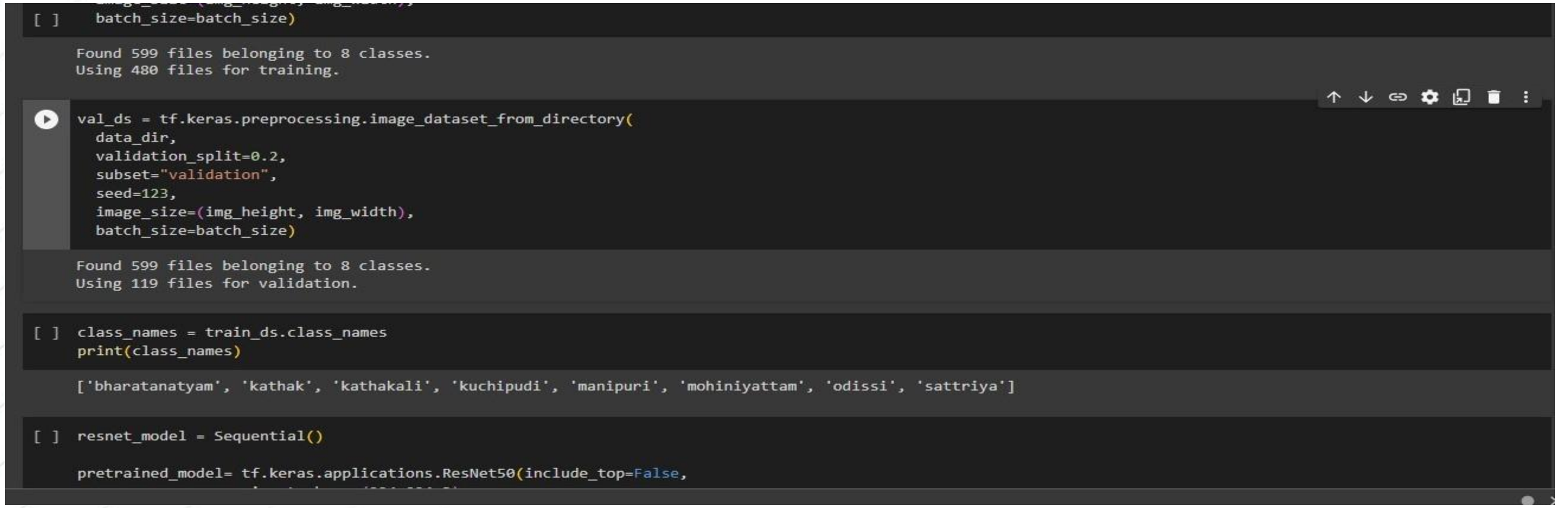
Confusion Matrix

bharatanatyam	61.5%	7.7%	0.0%	0.0%	0.0%	7.7%	15.4%	7.7%
kathak	0.0%	88.2%	0.0%	11.8%	0.0%	0.0%	0.0%	0.0%
kathakali	0.0%	0.0%	81.8%	0.0%	0.0%	18.2%	0.0%	0.0%
kuchipudi	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%
manipuri	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%
mohiniyattam	0.0%	0.0%	0.0%	6.2%	0.0%	93.8%	0.0%	0.0%

Training Model

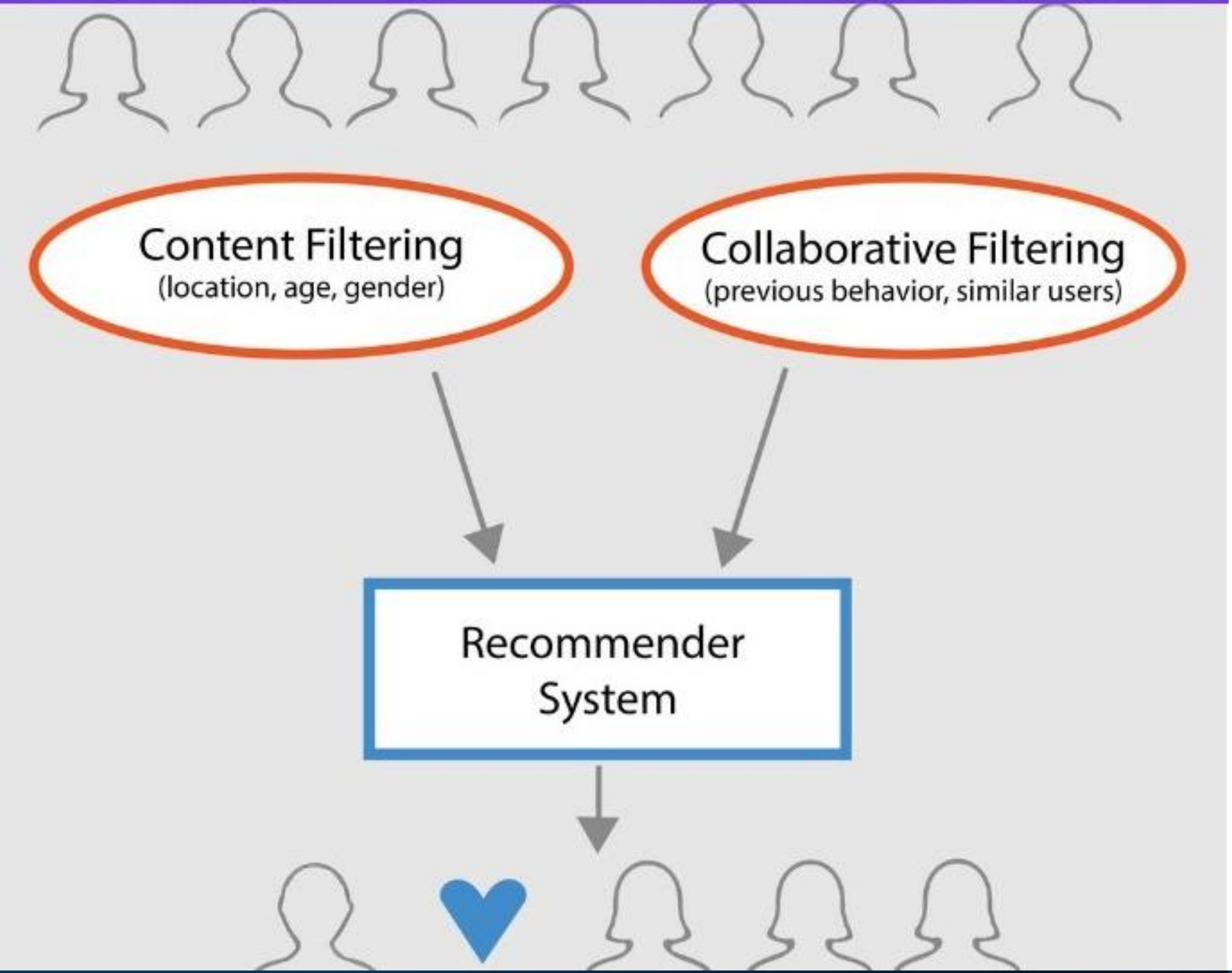


```
[ ]  model = keras.models.load_model("/content/drive/MyDrive/Actor_project/Dance_classification")

[ ]  predictions = []
     true_values = []
     for images , labels in val_ds:
       predictions.append(np.argmax(model.predict(images) , axis=1))
       true_values.append(labels)
     predictions_tensor = [tf.constant(array) for array in predictions]
     predictions_tensor = tf.concat(predictions_tensor , axis=0)
     true_values_tensor = tf.concat(true_values , axis=0)


     confusion_matrix = tf.math.confusion_matrix(labels=true_values_tensor , predictions=predictions_tensor)
     confusion_matrix  = confusion_matrix.numpy()

     1/1 [==============================] - 2s 2s/step
     1/1 [==============================] - 0s 45ms/step
     1/1 [==============================] - 0s 37ms/step
     1/1 [==============================] - 1s 1s/step
```



```
[ ]    batch_size=batch_size)

     Found 599 files belonging to 8 classes.
     Using 480 files for training.

     val_ds = tf.keras.preprocessing.image_dataset_from_directory(
       data_dir,
       validation_split=0.2,
       subset="validation",
       seed=123,
       image_size=(img_height, img_width),
       batch_size=batch_size)

     Found 599 files belonging to 8 classes.
     Using 119 files for validation.

[ ]  class_names = train_ds.class_names
     print(class_names)

     ['bharatanatyam', 'kathak', 'kathakali', 'kuchipudi', 'manipuri', 'mohiniyattam', 'odissi', 'sattriya']

[ ]  resnet_model = Sequential()

     pretrained_model= tf.keras.applications.ResNet50(include_top=False,
```

Classes



System Overview Diagram

Methodology

- Filtering Similar role
- Machine learning algorithms to compare dance movements
- Training and Dataset Preparation

Technologies

- Machine learning – Python
- React to front end
- Mongo DB
- Media Pipeline
- Cosine Similarity
- CNN(convolutional neural networks)
- ResNet

Research Gap Summary

Feature	A	B	C	D	Proposed S ystem
Dancer roll classification with motion Detection.	✗	✓	✓	✗	✓
Based On Cosine Similarity	✓	✗	✓	✓	✓
Research on privacy and security	✓	✓	✗	✗	✓
Diverse user demographics and skill levels	✗	✓	✗	✓	✓

Requirements

Functional Requirements:

- ❖ Ability to Recognize Movement features from video content
- ❖ Ability to classify character roles and dancing category from video content
- ❖ Accuracy – Classify dancers for most suitable category.

REFERENCES

GeeksforGeeks. (2022). *How to Calculate Cosine Similarity in Python?* [online] Available at: https://www.geeksforgeeks.org/how-to-calculate-cosine-similarity-in-python/ [Accessed 24 May 2023].

scikitlearn.org. (n.d.). *Loading...* [online] Available at: https://scikitlearn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html [Accessed 1 Aug. 2023].

www.sciencedirect.com. (n.d.). *Filtering Operation - an overview | ScienceDirect Topics*. [online] Available at: https://www.sciencedirect.com/topics/computer-science/filtering-operation [Accessed 1 Aug. 2023].

GeeksforGeeks. (2022). *How to Calculate Cosine Similarity in Python?* [online] Available at: https://www.geeksforgeeks.org/how-to-calculate-cosine-similarity-in-python/ [Accessed 24 May 2023].

Rainie, L. and Duggan, M. (2016). *Privacy and Information Sharing*. [online] Pew Research Center: Internet, Science & Tech. Available at: https://www.pewresearch.org/internet/2016/01/14/privacy-and-information-sharing/.

IT20050962 | M.A.R.F.M FAZIL

Classifying roles for Acting and Job recommendation

Bachelor of Science (Hons) in Information Technology Specializing in Interactive Media

Background

- ❖ Difficulty in Finding Talents for Acting
- ❖ Lengthy and Manual Casting Process
- ❖ Difficulty in Finding Opportunities for Acting
- ❖ Lack of Collaboration among Aspiring Creators

Research Gap Summery

Feature	A [1]	B [2]	C [3]	D [4]	Proposed System
Actor roll classification with emotion Recognition	✗	✓	✗	✗	✓
Based on CNN	✗	✗	✓	✗	✓
Face Recognition and characteristics Id entification	✗	✓	✓	✓	✓
Job Recommendation For actors using Image/Video based Approach	✗	✗	✗	✗	✓

Objectives

Specific Objective: Classifying roles for Acting and Job recommendation

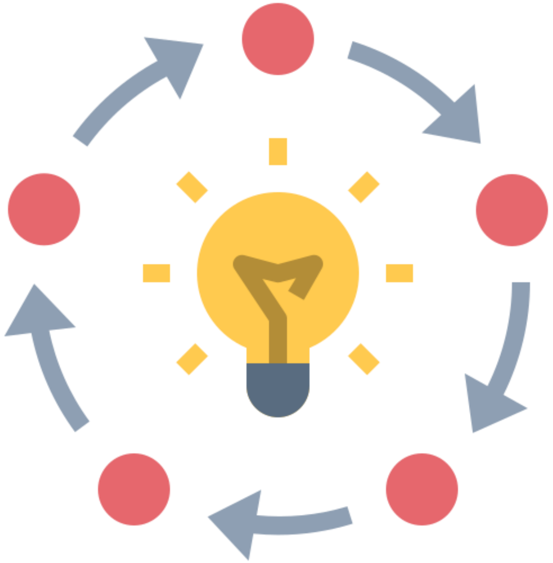
Specific Objective:

- ❖ Actor Role Classification with Emotion Recognition:
- ❖ Job Recommendation for Actors using Image/Video-Based Approach
- ❖ Advance machine learning Algorithms
- ❖ Enhancing Casting Process Efficiency

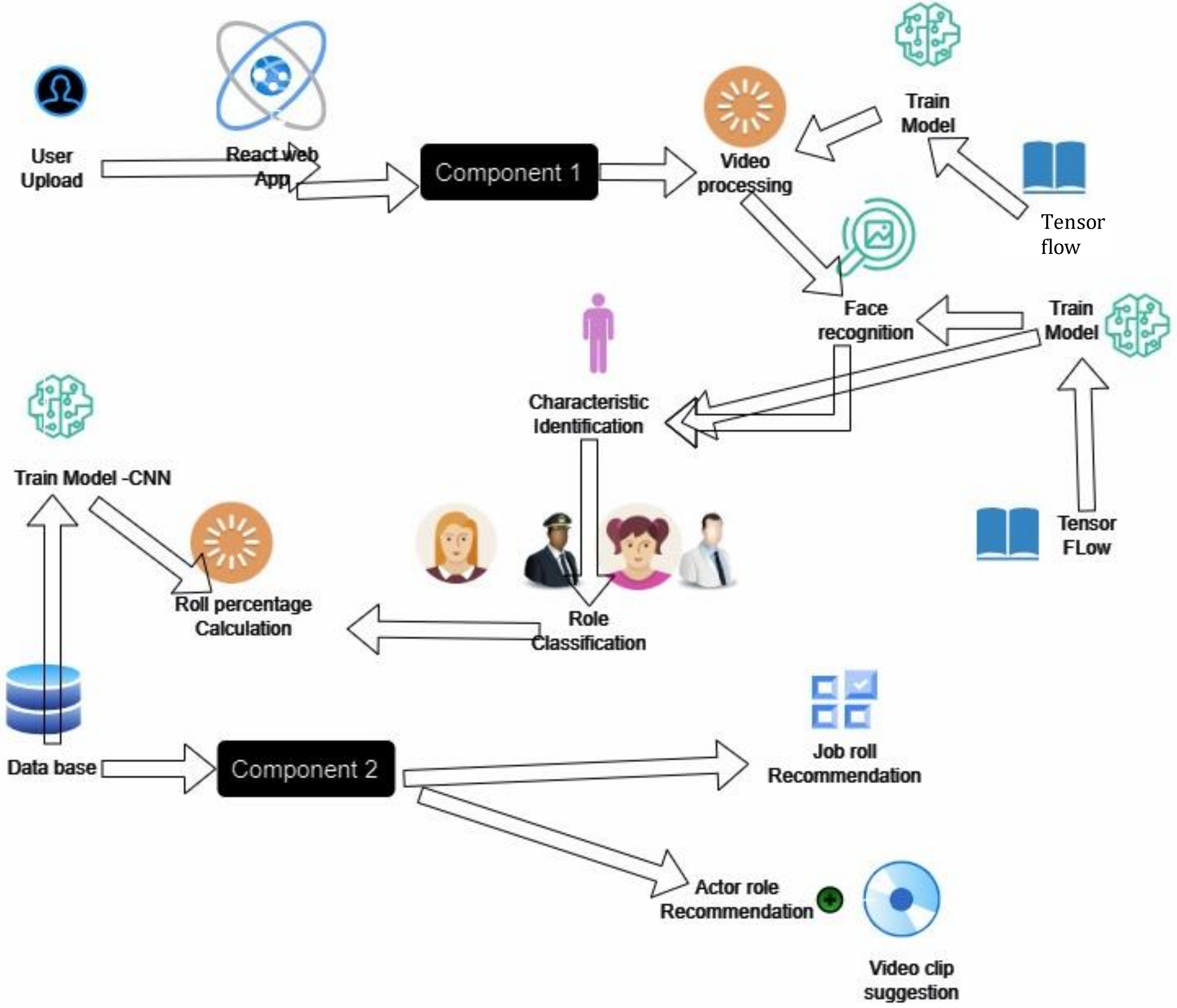
Research Questions

- why we recognize human with facial attributes?
- How to Classify Character roles ?
- How to Recommend Actors for a job post?
- How to recommend Job post for an Actor?

Methodology



System Overview Diagram



Methodology

- ❖ Gather the data for Actor image classification
- ❖ Preprocess the data by resizing and normalizing images.
- ❖ Design a Convolutional Neural Network (CNN) architecture.
- ❖ Train the model using the preprocessed dataset.

Data Collection

Data Pre-processing

Model Training

Data Visualization

Use the
Data results for recomme
ndation

Tools and Technologies

Technologies:

- CNN-VGG19

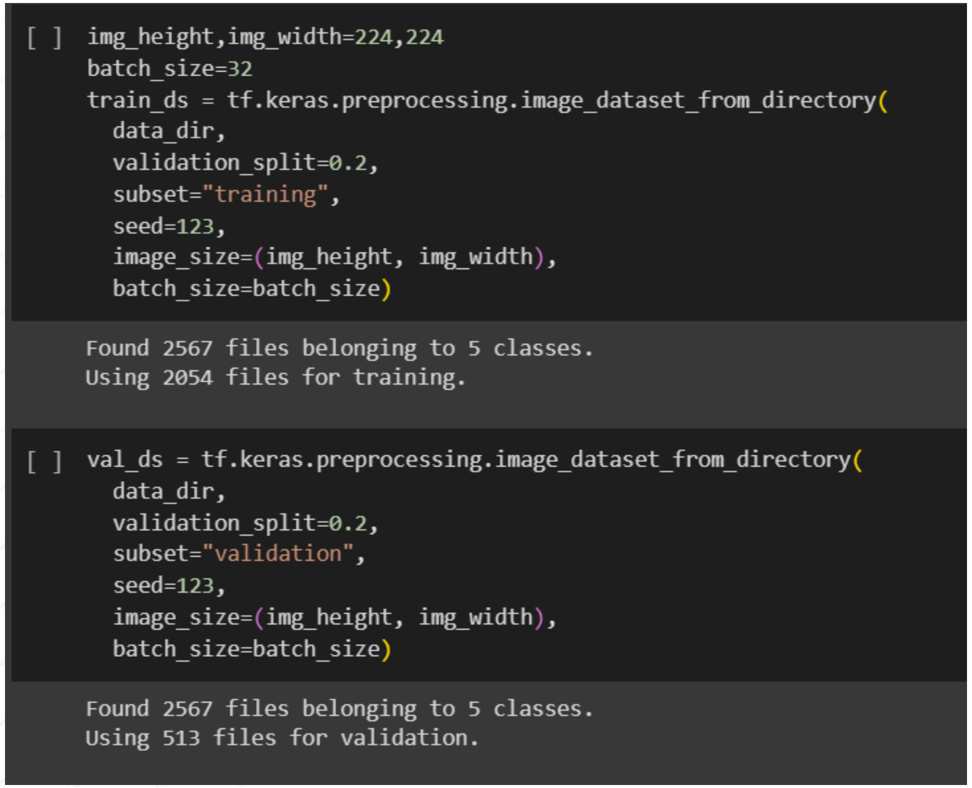
Tools:

- Machine learning – Python
- Hosting and Deploy – AWS
- For implementation needs Google collab
- Front end – React JS
- Database - mongo DB
- Fast API

Specific Objective:

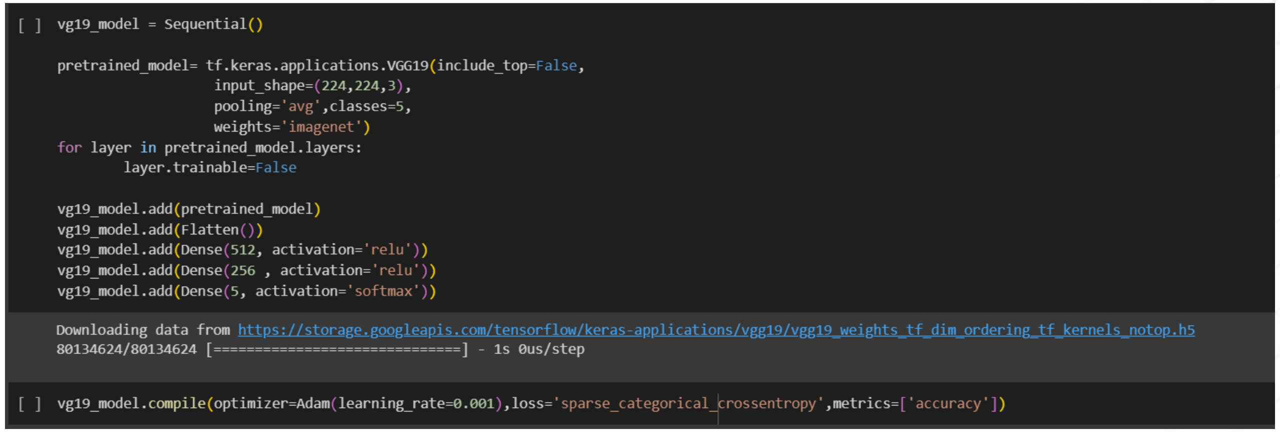
- Keras
- Face recognition and features – Tensor flow

Data Collection and Pre-processing

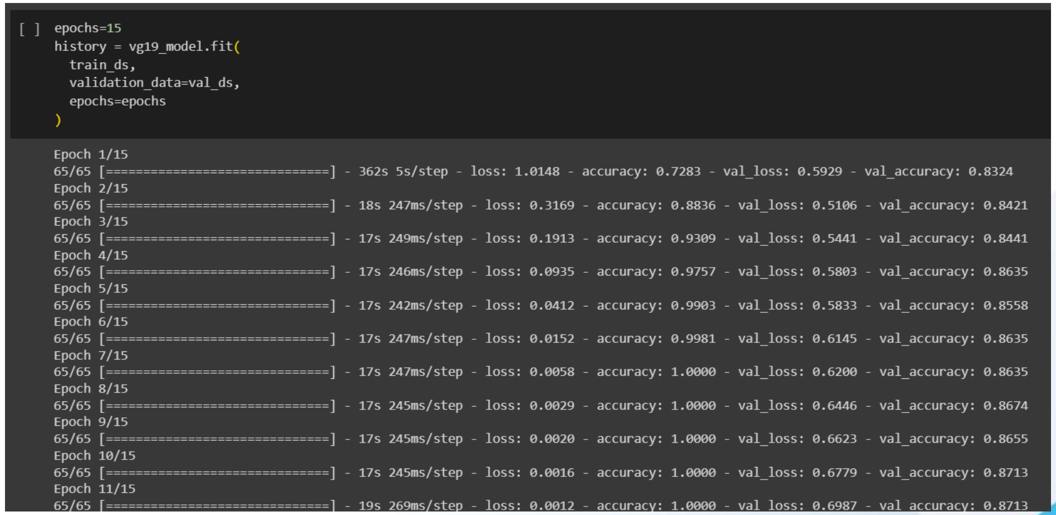


```
[ ] img_height,img_width=224,224
    batch_size=32
    train_ds = tf.keras.preprocessing.image_dataset_from_directory(
      data_dir,
      validation_split=0.2,
      subset="training",
      seed=123,
      image_size=(img_height, img_width),
      batch_size=batch_size)

    Found 2567 files belonging to 5 classes.
    Using 2054 files for training.

[ ] val_ds = tf.keras.preprocessing.image_dataset_from_directory(
      data_dir,
      validation_split=0.2,
      subset="validation",
      seed=123,
      image_size=(img_height, img_width),
      batch_size=batch_size)

    Found 2567 files belonging to 5 classes.
    Using 513 files for validation.
```

Model Training

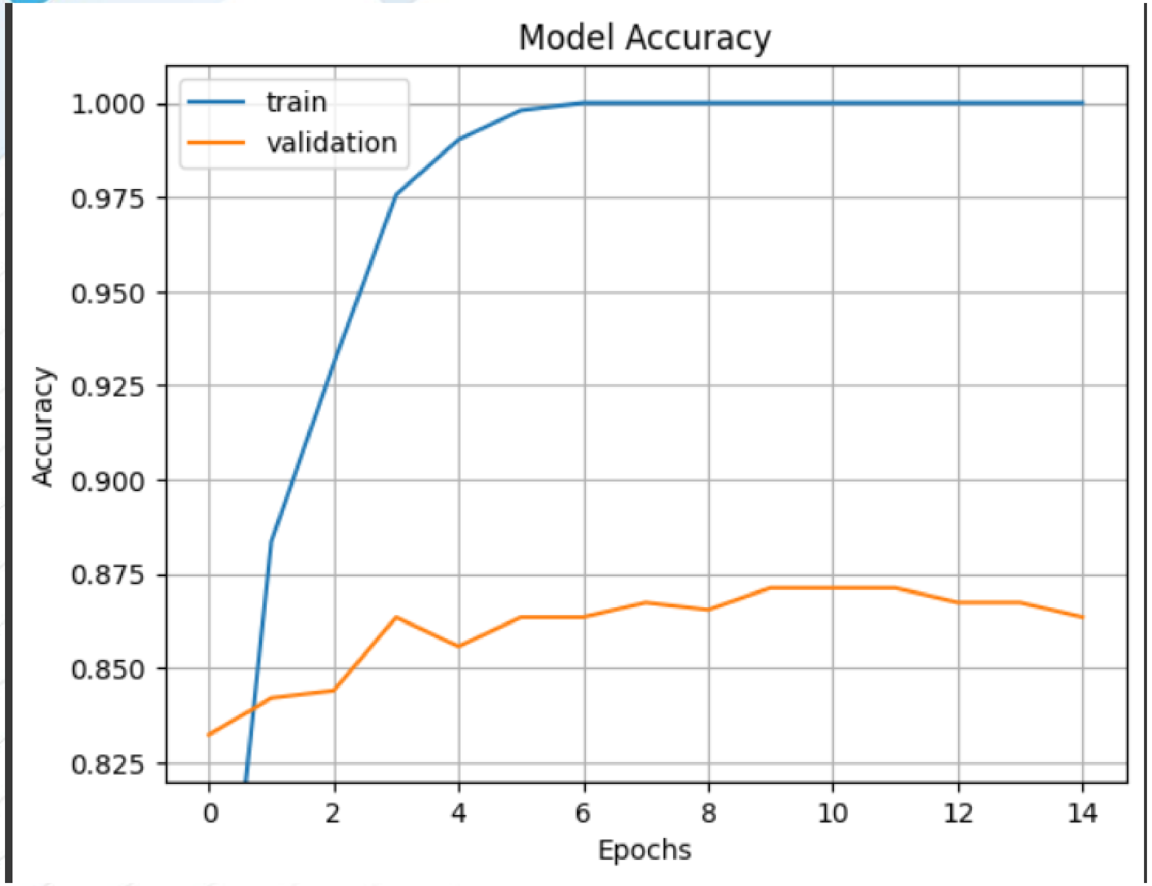


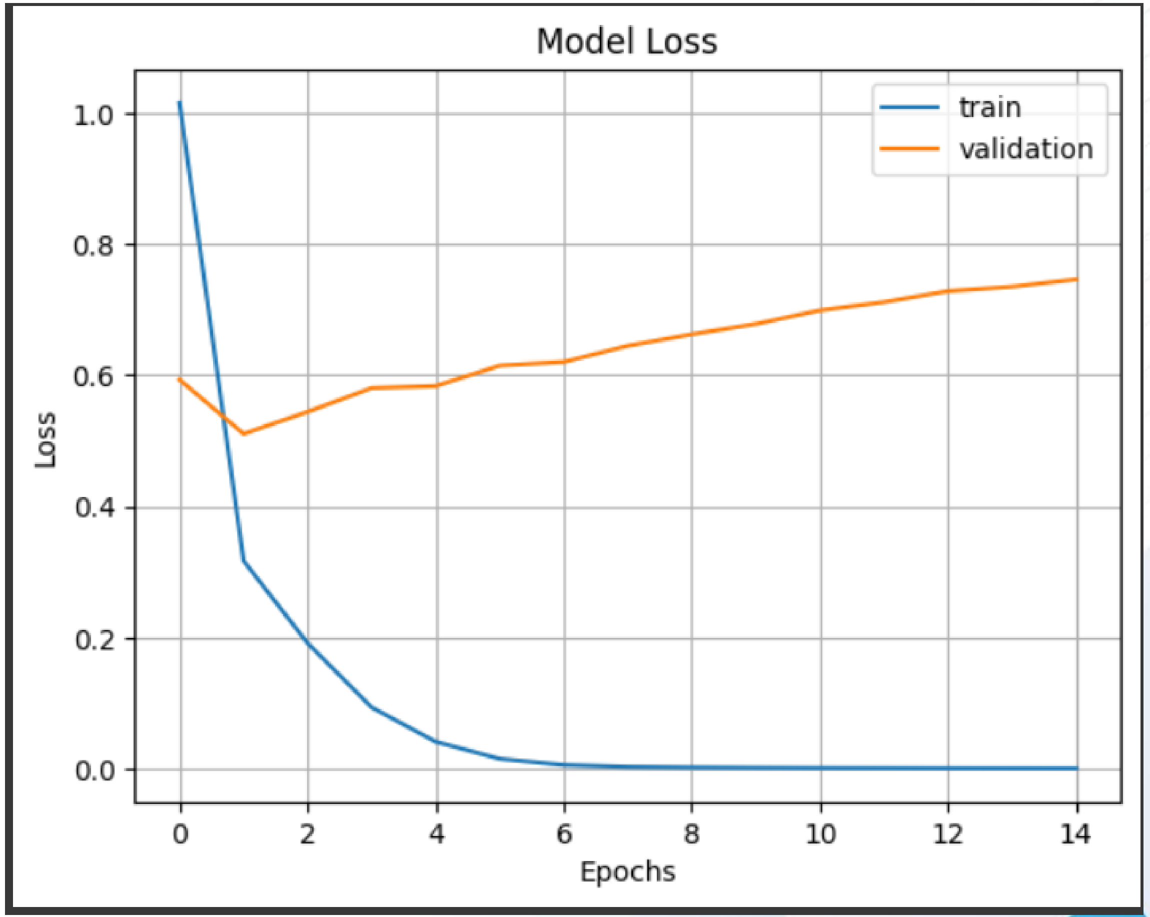
```
[ ] vg19_model = Sequential()

    pretrained_model= tf.keras.applications.VGG19(include_top=False,
                       input_shape=(224,224,3),
                       pooling='avg',classes=5,
                       weights='imagenet')
    for layer in pretrained_model.layers:
            layer.trainable=False

    vg19_model.add(pretrained_model)
    vg19_model.add(Flatten())
    vg19_model.add(Dense(512, activation='relu'))
    vg19_model.add(Dense(256 , activation='relu'))
    vg19_model.add(Dense(5, activation='softmax'))

    Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
    80134624/80134624 [==============================] - 1s 0us/step

[ ] vg19_model.compile(optimizer=Adam(learning_rate=0.001),loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

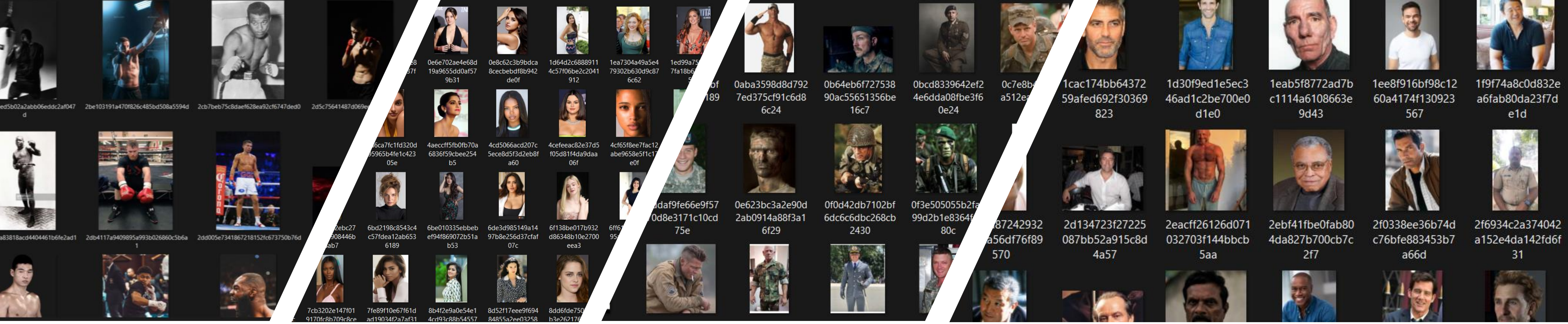


```
[ ] epochs=15
    history = vg19_model.fit(
      train_ds,
      validation_data=val_ds,
      epochs=epochs
    )

    Epoch 1/15
    65/65 [==============================] - 362s 5s/step - loss: 1.0148 - accuracy: 0.7283 - val_loss: 0.5929 - val_accuracy: 0.8324
    Epoch 2/15
    65/65 [==============================] - 18s 247ms/step - loss: 0.3169 - accuracy: 0.8836 - val_loss: 0.5106 - val_accuracy: 0.8421
    Epoch 3/15
    65/65 [==============================] - 17s 249ms/step - loss: 0.1913 - accuracy: 0.9309 - val_loss: 0.5441 - val_accuracy: 0.8441
    Epoch 4/15
    65/65 [==============================] - 17s 246ms/step - loss: 0.0935 - accuracy: 0.9757 - val_loss: 0.5803 - val_accuracy: 0.8635
    Epoch 5/15
    65/65 [==============================] - 17s 242ms/step - loss: 0.0412 - accuracy: 0.9903 - val_loss: 0.5833 - val_accuracy: 0.8558
    Epoch 6/15
    65/65 [==============================] - 17s 247ms/step - loss: 0.0152 - accuracy: 0.9981 - val_loss: 0.6145 - val_accuracy: 0.8635
    Epoch 7/15
    65/65 [==============================] - 17s 247ms/step - loss: 0.0058 - accuracy: 1.0000 - val_loss: 0.6200 - val_accuracy: 0.8635
    Epoch 8/15
    65/65 [==============================] - 17s 245ms/step - loss: 0.0029 - accuracy: 1.0000 - val_loss: 0.6446 - val_accuracy: 0.8674
    Epoch 9/15
    65/65 [==============================] - 17s 245ms/step - loss: 0.0020 - accuracy: 1.0000 - val_loss: 0.6623 - val_accuracy: 0.8655
    Epoch 10/15
    65/65 [==============================] - 17s 245ms/step - loss: 0.0016 - accuracy: 1.0000 - val_loss: 0.6779 - val_accuracy: 0.8713
    Epoch 11/15
    65/65 [==============================] - 19s 269ms/step - loss: 0.0012 - accuracy: 1.0000 - val_loss: 0.6987 - val_accuracy: 0.8713
```

CNN ARCHITECTURES EVIDENCE OF COMPLETION






2be103191a470f826c485bd508a5594d
2cb7beb75c8daef628ea92cf6747ded0
2db4117a9409895a993b026860c5b6a1
2dd005e7341867218152fc673750b76d
0e6e702ae4e68d19a9655dd0af579b31
0e8c62c3b9bdca8cecbebdf8b942de0f
1d64d2c68889114c57f06be2c2041912
1ea7304a49a5e479302b630d9c876c62
4aeccff5fb0fb70a6836f59cbee254b5
4cd5066acd207c5ece8d5f3d2eb8fa60
4cefeeac82e37d5f05d81f4da9daa06f
4cf65f8ee7fac12abe9658e5f1c1e0f
6bd2198c8543c4c57fdea12ab6536189
6be010335ebbebef94f869072b51ab53
6de3d985149a1497b8e256d37cfaf07c
6f138be017b932d86348b10e2700eea3
0aba3598d8d7927ed375cf91c6d86c24
0b64eb6f72753890ac55651356be16c7
0bcd8339642ef24e6dda08fbe3f60e24
0e623bc3a2e90d2ab0914a88f3a16f29
0f0d42db7102bf6dc6c6dbc268cb2430
1cac174bb6437259afed692f30369823
1d30f9ed1e5ec346ad1c2be700e0d1e0
1eab5f8772ad7bc1114a6108663e9d43
1ee8f916bf98c1260a4174f130923567
1f9f74a8c0d832ea6fab80da23f7de1d
2d134723f27225087bb52a915c8d4a57
2eacff26126d071032703f144bbcb5aa
2ebf41fbe0fab804da827b700cb7c2f7
2f0338ee36b74dc76bfe883453b7a66d
2f6934c2a374042a152e4da142fd6f31

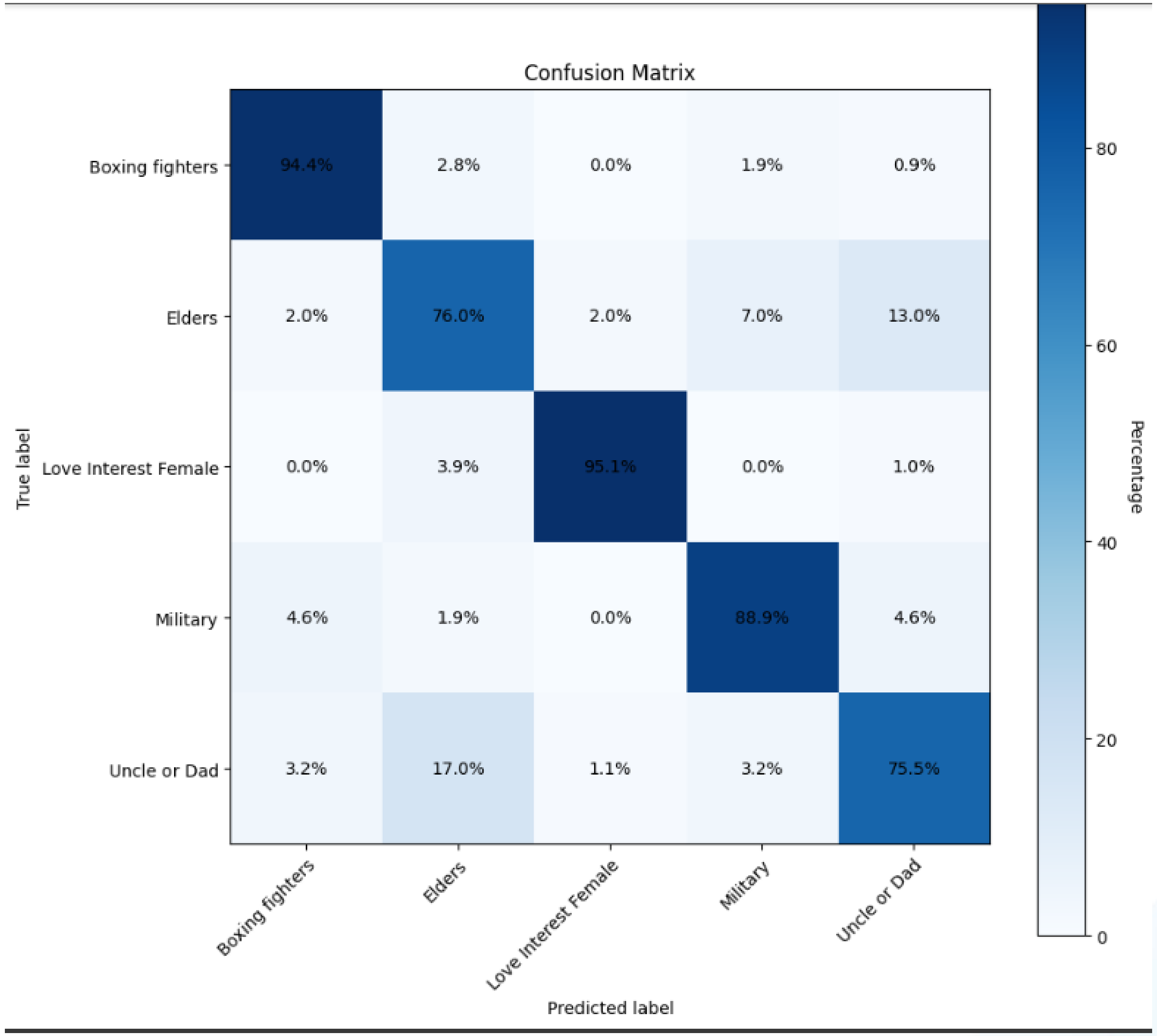
CNN ARCHITECTURES EVIDENCE OF COMPLETION

sample test images

0b1dda72a692d8f098f95f64c6b50526
0ed8400238f7c82165aff2c0c7d5fabc
0f325b2b0f9c9042c2fc5dad58573176
0f429509155d7e98f4dea1e53c77bb80
1.500_F_165463047_g0VKYO48Gf8fPpymiyiVEQu0s9fgXIA4
1.646791d9-900d-44f5-852a-7c01733c4021
1.fa5a99b4dc8181c8aac9a3246f0f8be0
1.happy-senior-elderly-couple-old-people-selfie
1.old-indian-female-and
1e5dd92490fee7b2f4335e794a4d
2.0d7b048a6f33790ffb5e6b9ac896
2.61d5Q%2BTiWNL_UX385
2.500_F_165463032_PKs3x67sqT96
2.619-01721726en_Masterfile
2.630-03479591em-closeup-of-a-s

IT20050962 | Fazil |

Data Visualization



Completion and Future Works

50%

Completion of the components

❖ Collecting real test dataset with variety

❖ Model Training and Prediction

100

Future Implements

❖ Train another Model for analyze Accuracy

❖ Frontend Implementation

Requirements

Functional Requirements:

- ❖ Ability to Recognize facial features from video content
- ❖ Ability to classify character roles from video content

Functional Requirements:

- ❖ Accuracy – Classify characters for most suitable roles
- ❖ Usability – understand users with different roles on his/her videos
- ❖ Performance - Fast loading results without any lagging on website

REFERENCES

A[1] A.-N. Lee, K.-Y. C. and C.-T. L. , "ActRec: A Word Embedding-based Approach to Recommend Movie Actors to Match Role Descriptions," IEEE, The Hague, Netherlands, 2020.

B[2] H. X. and C. L. , "Face Recognition and Application of Film and Television Actors Based on Dlib," IEEE, Suzhou, China, 2019.

C[3] A. K. and C. T. , "Using CNN and OpenCV, Emotion Recognition with Facial Feature Approach," IEEE, Greater Noida, India, 2023.

D[4] T. C. . M. D. . M. v. d. G. C. H. and H. K. , "Multimodal Emotion Recognition for Visualizing," ICT, LEIDEN, THE NETHERLANDS, 2020.