

# Identifying Lead Cast In Movies Using Transfer Learning Models

*Submitted by -  
Md. Musharraf  
Md. Hamid  
Rashid Aziz*

*Guided by-  
Dr. Tanvir Ahmad*

# I n d e x

Objective

Motivation

Methodology

Screenshots

Results

Technology Used

Conclusion

## Objective

Detecting and naming actors in movies are important for content-based indexing and retrieval of movie scenes and can also be used to support statistical analysis of the film style.

Detecting and naming actors in unedited footage can be useful for post-production.

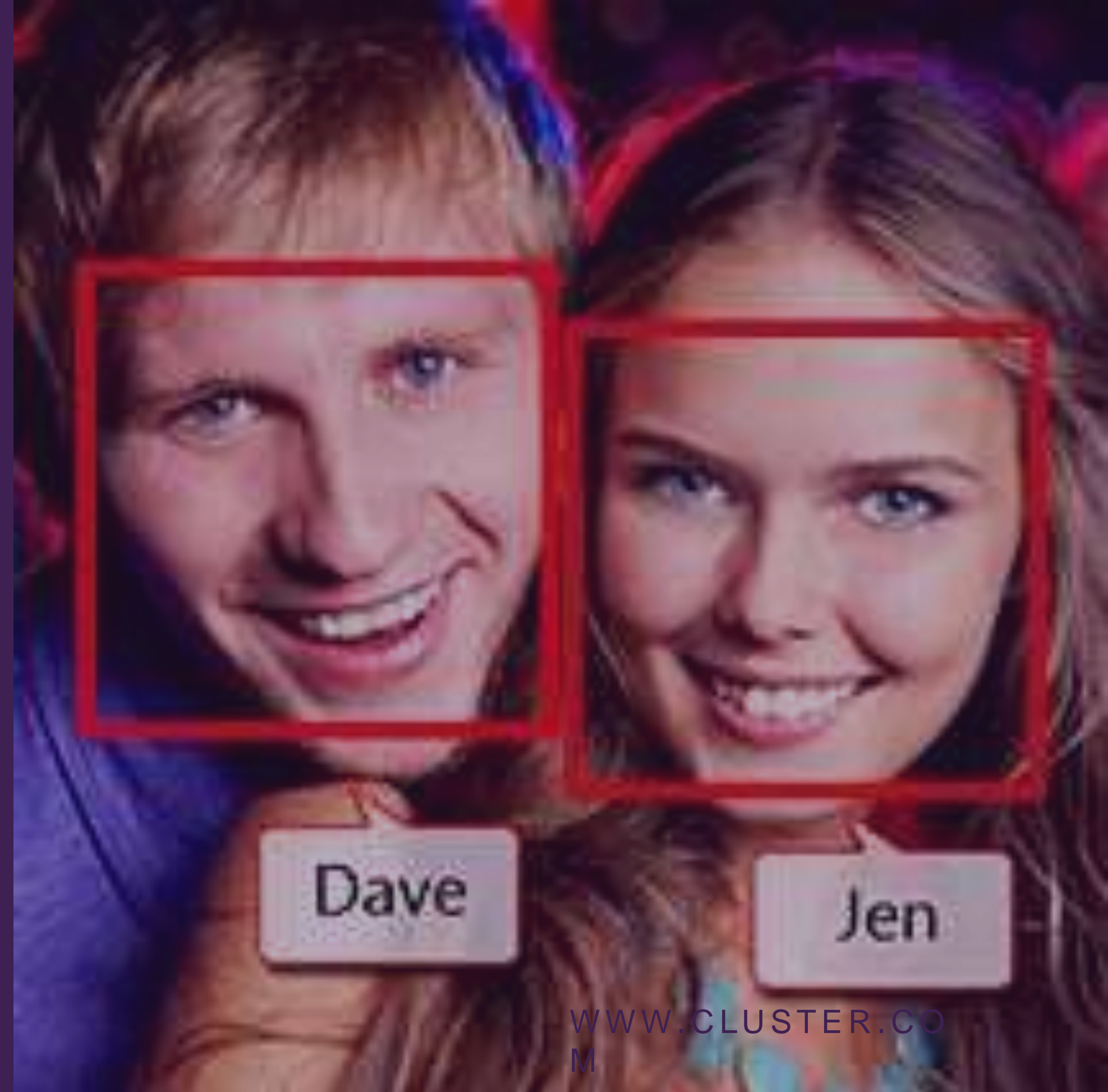




# MOTIVATION

---

Recognizing human faces in wild is emerging as a critically important and technically challenging computer vision problem. With a few notable exceptions, most previous works in the last several decades have focused on recognizing faces captured in a laboratory setting. However, with the introduction of databases, face recognition community is gradually shifting its focus on much more challenging unconstrained settings.



# PROPOSED METHOD

1. We plan to apply Transfer Learning models of Tensorflow to cut down the computation power required.
2. We will firstly prune the dataset, deleting those image that doesn't contain a recognizable human face then train the dataset of actors/actress using the ImageNet model.
3. When training is done we will input the film and extract the frames every 36 seconds. After extracting is done we will again apply face detection to delete those image that either doesn't have a face or have blurry face.
4. After all the pruning work is done we will apply our Tensorflow model



# Screenshots

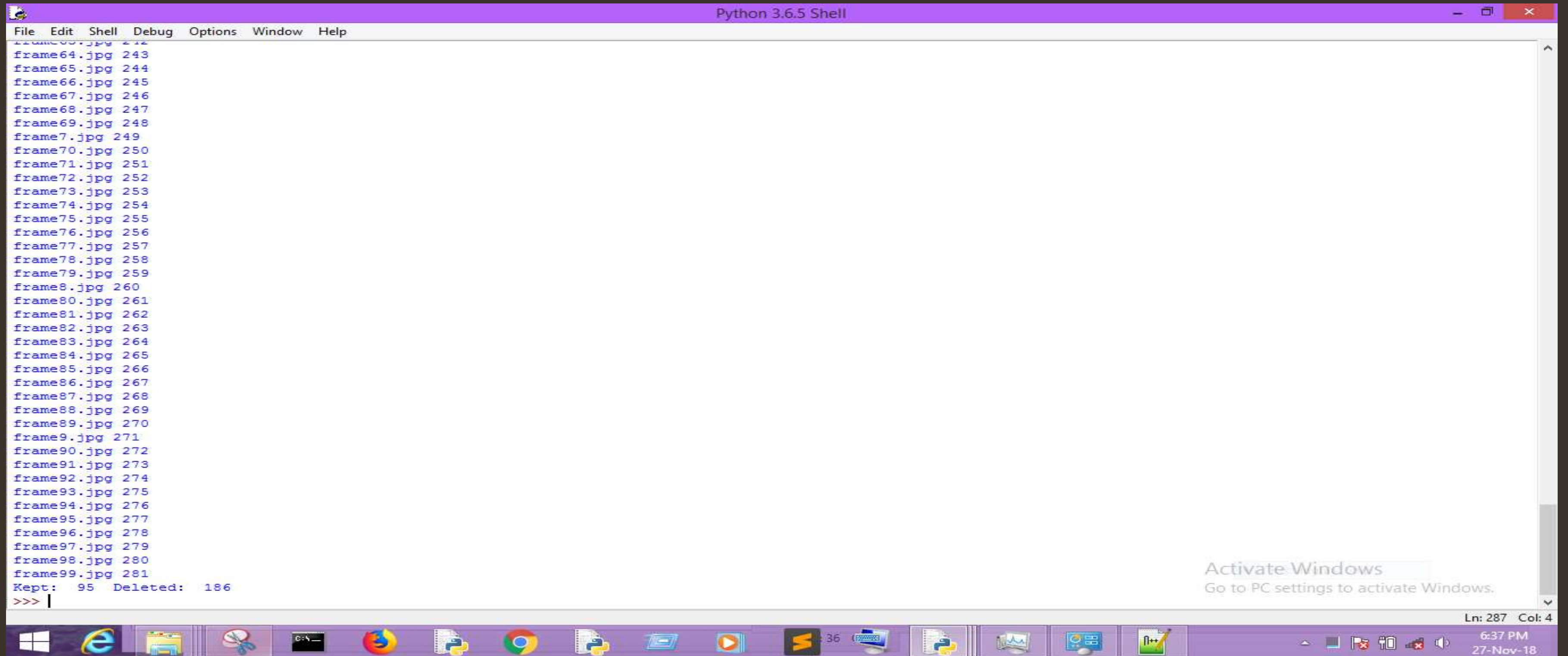
## Creation of Bottlenecks

```
/usr/bin/bash --login -i E:\projects\minor\codes\final codes\train.sh

Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-873092890-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-874729890-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-887939388-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-887939394-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-887939942-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-887940262-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-887941126-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-895183692-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927403932-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927404676-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927404686-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927404692-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927409752-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927409760-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927424514-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927424538-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927424562-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927424570-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927424588-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927425650-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927470550-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927500786-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927500836-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-927500856-612x612.jpg.txt
1000 bottleneck files created.
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1025256198-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1025365232-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1063410392-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1063428706-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1063428708-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1063428710-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1063441782-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1063442034-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1069436118-612x612.jpg.txt
Creating bottleneck at tf_files/bottlenecks\Emma Stone\gettyimages-1069436236-612x612.jpg.txt
1100 bottleneck files created.
```

# Screenshots

## Removing No Face/ Blurry images



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
frame64.jpg 243
frame65.jpg 244
frame66.jpg 245
frame67.jpg 246
frame68.jpg 247
frame69.jpg 248
frame7.jpg 249
frame70.jpg 250
frame71.jpg 251
frame72.jpg 252
frame73.jpg 253
frame74.jpg 254
frame75.jpg 255
frame76.jpg 256
frame77.jpg 257
frame78.jpg 258
frame79.jpg 259
frame8.jpg 260
frame80.jpg 261
frame81.jpg 262
frame82.jpg 263
frame83.jpg 264
frame84.jpg 265
frame85.jpg 266
frame86.jpg 267
frame87.jpg 268
frame88.jpg 269
frame89.jpg 270
frame9.jpg 271
frame90.jpg 272
frame91.jpg 273
frame92.jpg 274
frame93.jpg 275
frame94.jpg 276
frame95.jpg 277
frame96.jpg 278
frame97.jpg 279
frame98.jpg 280
frame99.jpg 281
Kept: 95 Deleted: 186
>>> |
```

Activate Windows  
Go to PC settings to activate Windows.

Ln: 287 Col: 4  
6:37 PM  
27-Nov-18



# Screenshots

## Training and Testing Accuracy

Training Accuracy: 80.00%

Validation Accuracy: 74.00%

Cross-Entropy: 0.1953

Final Test Accuracy: 65.9%

```
/usr/bin/bash --login -i E:\projects\minor\codes\final codes\train.sh
2018-11-27 17:52:12.458250: Step 350: Cross entropy = 1.284309
2018-11-27 17:52:12.661383: Step 350: Validation accuracy = 68.0% (N=100)
2018-11-27 17:52:14.601580: Step 360: Train accuracy = 68.0%
2018-11-27 17:52:14.601580: Step 360: Cross entropy = 1.219738
2018-11-27 17:52:14.789093: Step 360: Validation accuracy = 72.0% (N=100)
2018-11-27 17:52:16.720242: Step 370: Train accuracy = 71.0%
2018-11-27 17:52:16.720242: Step 370: Cross entropy = 1.228932
2018-11-27 17:52:16.907751: Step 370: Validation accuracy = 66.0% (N=100)
2018-11-27 17:52:19.871899: Step 380: Train accuracy = 67.0%
2018-11-27 17:52:19.871899: Step 380: Cross entropy = 1.234554
2018-11-27 17:52:20.177439: Step 380: Validation accuracy = 60.0% (N=100)
2018-11-27 17:52:22.151474: Step 390: Train accuracy = 67.0%
2018-11-27 17:52:22.151474: Step 390: Cross entropy = 1.199238
2018-11-27 17:52:22.338987: Step 390: Validation accuracy = 60.0% (N=100)
2018-11-27 17:52:24.278776: Step 400: Train accuracy = 64.0%
2018-11-27 17:52:24.278776: Step 400: Cross entropy = 1.167836
2018-11-27 17:52:24.466287: Step 400: Validation accuracy = 64.0% (N=100)
2018-11-27 17:52:27.147900: Step 410: Train accuracy = 68.0%
2018-11-27 17:52:27.147900: Step 410: Cross entropy = 1.235429
2018-11-27 17:52:27.476043: Step 410: Validation accuracy = 71.0% (N=100)
2018-11-27 17:52:29.742751: Step 420: Train accuracy = 71.0%
2018-11-27 17:52:29.742751: Step 420: Cross entropy = 1.191243
2018-11-27 17:52:29.945885: Step 420: Validation accuracy = 64.0% (N=100)
2018-11-27 17:52:31.887888: Step 430: Train accuracy = 82.0%
2018-11-27 17:52:31.887888: Step 430: Cross entropy = 1.001163
2018-11-27 17:52:32.087172: Step 430: Validation accuracy = 68.0% (N=100)
2018-11-27 17:52:34.500551: Step 440: Train accuracy = 63.0%
2018-11-27 17:52:34.500551: Step 440: Cross entropy = 1.182885
2018-11-27 17:52:34.781810: Step 440: Validation accuracy = 64.0% (N=100)
2018-11-27 17:52:37.261160: Step 450: Train accuracy = 75.0%
2018-11-27 17:52:37.261160: Step 450: Cross entropy = 1.145484
2018-11-27 17:52:37.448670: Step 450: Validation accuracy = 58.0% (N=100)
2018-11-27 17:52:39.384046: Step 460: Train accuracy = 71.0%
2018-11-27 17:52:39.384046: Step 460: Cross entropy = 1.138090
2018-11-27 17:52:39.571556: Step 460: Validation accuracy = 69.0% (N=100)
2018-11-27 17:52:41.762390: Step 470: Train accuracy = 71.0%
2018-11-27 17:52:41.762390: Step 470: Cross entropy = 1.061785
2018-11-27 17:52:42.043658: Step 470: Validation accuracy = 66.0% (N=100)
2018-11-27 17:52:44.795278: Step 480: Train accuracy = 75.0%
2018-11-27 17:52:44.795278: Step 480: Cross entropy = 1.125413
2018-11-27 17:52:44.982788: Step 480: Validation accuracy = 70.0% (N=100)
2018-11-27 17:52:46.993850: Step 490: Train accuracy = 78.0%
2018-11-27 17:52:46.993850: Step 490: Cross entropy = 1.009236
2018-11-27 17:52:47.182796: Step 490: Validation accuracy = 72.0% (N=100)
2018-11-27 17:52:48.931829: Step 499: Train accuracy = 80.0%
2018-11-27 17:52:48.931829: Step 499: Cross entropy = 0.976891
2018-11-27 17:52:49.141926: Step 499: Validation accuracy = 74.0% (N=100)
Final test accuracy = 65.9% (N=458)
Converted 2 variables to const ops.
Training finished
```



# Screenshots

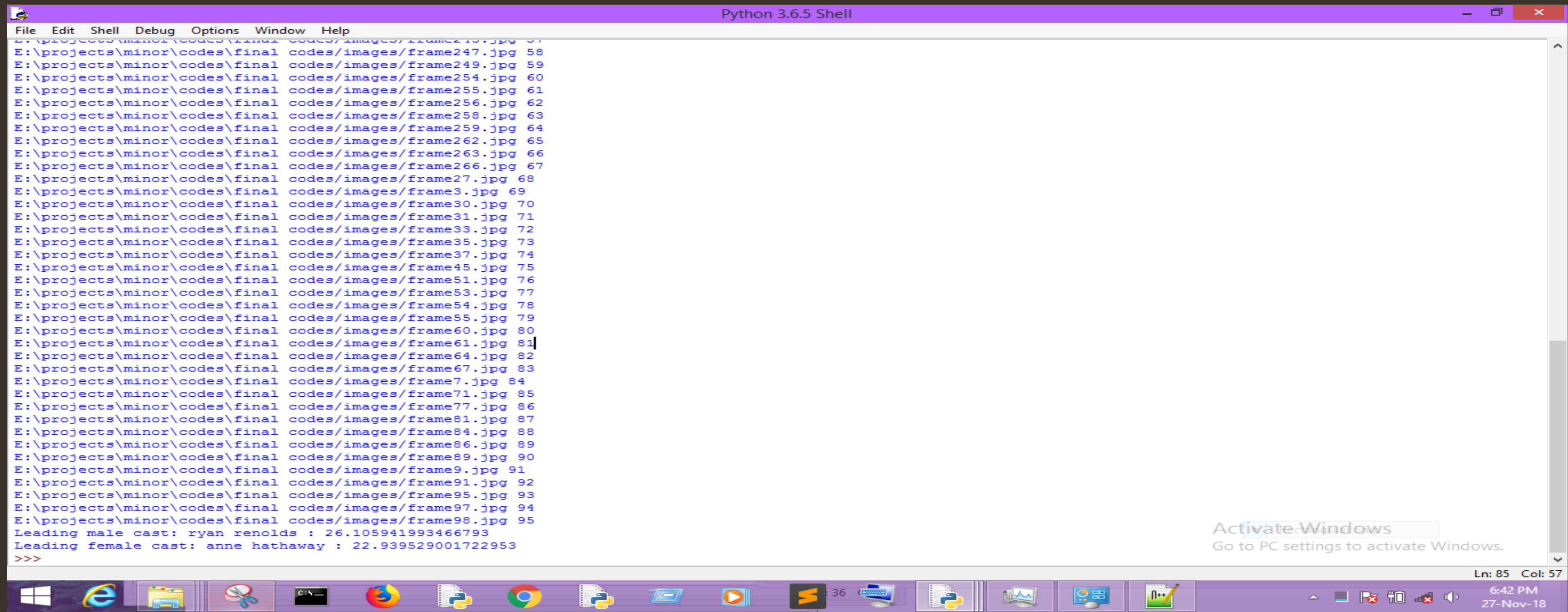
## Final Result (Input Film: Love and other drugs)

Leading Male Cast: Ryan Reynolds (error)

Correct Male Leading Cast: Jake Gyllenhaal

Leading female cast: Anne Hathaway (correct)

Correct Female Leading Cast: Anne Hathaway



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
E:\projects\minor\codes\final codes/images/frame247.jpg 58
E:\projects\minor\codes\final codes/images/frame249.jpg 59
E:\projects\minor\codes\final codes/images/frame254.jpg 60
E:\projects\minor\codes\final codes/images/frame255.jpg 61
E:\projects\minor\codes\final codes/images/frame256.jpg 62
E:\projects\minor\codes\final codes/images/frame258.jpg 63
E:\projects\minor\codes\final codes/images/frame259.jpg 64
E:\projects\minor\codes\final codes/images/frame262.jpg 65
E:\projects\minor\codes\final codes/images/frame263.jpg 66
E:\projects\minor\codes\final codes/images/frame266.jpg 67
E:\projects\minor\codes\final codes/images/frame27.jpg 68
E:\projects\minor\codes\final codes/images/frame3.jpg 69
E:\projects\minor\codes\final codes/images/frame30.jpg 70
E:\projects\minor\codes\final codes/images/frame31.jpg 71
E:\projects\minor\codes\final codes/images/frame33.jpg 72
E:\projects\minor\codes\final codes/images/frame35.jpg 73
E:\projects\minor\codes\final codes/images/frame37.jpg 74
E:\projects\minor\codes\final codes/images/frame45.jpg 75
E:\projects\minor\codes\final codes/images/frame51.jpg 76
E:\projects\minor\codes\final codes/images/frame53.jpg 77
E:\projects\minor\codes\final codes/images/frame54.jpg 78
E:\projects\minor\codes\final codes/images/frame55.jpg 79
E:\projects\minor\codes\final codes/images/frame60.jpg 80
E:\projects\minor\codes\final codes/images/frame61.jpg 81
E:\projects\minor\codes\final codes/images/frame64.jpg 82
E:\projects\minor\codes\final codes/images/frame67.jpg 83
E:\projects\minor\codes\final codes/images/frame7.jpg 84
E:\projects\minor\codes\final codes/images/frame71.jpg 85
E:\projects\minor\codes\final codes/images/frame77.jpg 86
E:\projects\minor\codes\final codes/images/frame81.jpg 87
E:\projects\minor\codes\final codes/images/frame84.jpg 88
E:\projects\minor\codes\final codes/images/frame86.jpg 89
E:\projects\minor\codes\final codes/images/frame89.jpg 90
E:\projects\minor\codes\final codes/images/frame9.jpg 91
E:\projects\minor\codes\final codes/images/frame91.jpg 92
E:\projects\minor\codes\final codes/images/frame95.jpg 93
E:\projects\minor\codes\final codes/images/frame97.jpg 94
E:\projects\minor\codes\final codes/images/frame98.jpg 95
Leading male cast: ryan reynolds : 26.105941993466793
Leading female cast: anne hathaway : 22.939529001722953
>>>
```

Activate Windows  
Go to PC settings to activate Windows.

Ln: 85 Col: 57

6:42 PM  
27-Nov-18

Input Film: Love And Other Drugs

Correct Leading Cast: Jake Gyllenhaal and Anne Hathaway

Output:

Training Accuracy: 80.00%

Validation Accuracy: 74.00%

Final Test Accuracy: 65.9%

Cross-Entropy: 0.1953

Leading Cast: Ryan Reynolds and Anne Hathaway





# Technology Used

- Python
- Tensorflow
- Transfer Learning
- OpenCV







Thank You