INSTAGRAM PROFILE CLASSIFICATION USING DEEP LEARNING

CONTENT

- Introduction
- objective
- Tools
- Data
- EDA and Processing
- Modelling

INTRODUCTION

OBJECTIVE

- The object of this project is to build a deep neural network model that uses screenshots of several posts on an Instagram page profile as one image to identify the interest of that page

TOOLS

For manipulating and visualization:

Numpy

Pandas

Matplotlib

Jupyter

For modeling:

Scikit-Learn

TensorFlow + Keras

DATA

fitness

fitness

beauty

fitness

food

family

Dataset

- Contain a screenshots of several posts on an Instagram pages as one image into 5 different interest of that page.

Size

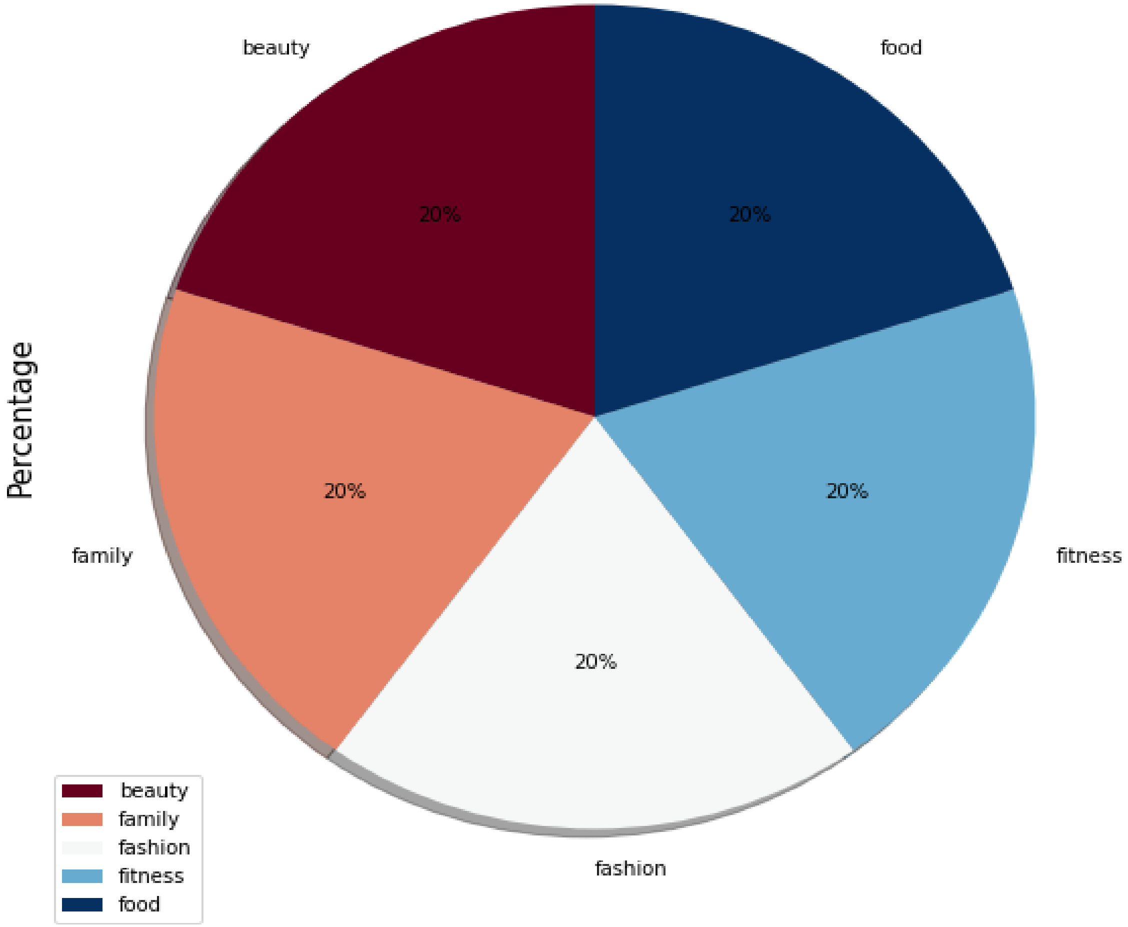
- 3770 images

Source

- From kaggle

EDA AND PROCESSING

- Distribution of images in each class.



- Processing
  - Resize images to height and width into 200 and 150
  - Normalization (divide by 255)

- Data augmentation
  - Flip ‘horizontal ‘and ‘vertical’
  - Rotation (90)
  - Shifting
  - zoom (0.2)

- Split Dataset
  - Train: 80% , 3053 images
  - Validation: 10%, 340 images
  - Test: 10%, 377 images

MODELING: BASELINE

MODEL	Parameters	Training accuracy	Validation accuracy
Logistic regression	C=10^(-4)	42.6%	42%
Simple NN	- Hidden layers= 1 - Neurons=25 - Activation = tanh - Output layer = 5 Neurons Activation= softmax - Epochs=30	74.32%	69.11%

MODELING: CONVOLUTIONAL NEURAL NETWORKS

Note:

- Epochs always assign for 30
- Early stopping=[monitor=‘val loss’, patience=1, mode=‘min]
- Output layers= [dense=5 , Activation = softmax]

Model	Convolutional layers	Hidden layers	Optimizer	Early stop	Accuracy	Loss
CNN 1	- Filters=[5,10,15] - Activation=tanh - Padding=same - MaxPooling2D	- Dense=[125,25] - Activation=tanh - Dropout = 0.5	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	6 epochs	- Training = 53.12% - Validation = 52.64%	- Training= 1.02 - Validation=1.03
CNN 2	- Filters=[5,10,15] - Activation=tanh - Padding=same - MaxPooling2D	- Dense=[250,125 ,25] - Activation=tanh - Dropout = 0.5	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	6 epochs	- Training = 55.81% - Validation = 55.52%	- Training =1.01 - Validation = 1.04
CNN 3	- Filters=[30,60,90] - Activation=tanh - Padding=same - MaxPooling2D	- Dense=[125,25] - Activation=tanh - Dropout = 0.05	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	7 epochs	- Training = 64.13% - Validation = 64.11%	- Training =0.85 - Validation = 0.9
CNN 4	- Filters=[30,60,90] - Activation=tanh - Padding=valid - MaxPooling2D	- Dense=[250,125 ,25] - Activation=tanh - Dropout = 0.05	- Adam (lr=0.01) - Loss = categorical crossentropy - Metrics = accuracy	10 epochs	- Training = 76.25% - Validation = 75.25%	- Training =0.58 - Validation = 0.6
CNN 5	- Filters=[30,30,30] - Activation=tanh - Padding=valid - MaxPooling2D	- Dense=[500,250 ,125,25] - Activation=tanh - Dropout = 0.5	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	8 epochs	- Training =77.39% - Validation = 76.29%	- Training =0.54 - Validation = 0.6

MODELING: TRANSFER LEARNING

Note:

- Epochs always assign for 30
- Early stopping=[monitor=‘val loss’, patience=1, mode=‘min]
- Output layers= [dense=5 , Activation = softmax]
- Base=[weights=imagenet, Include top= False]

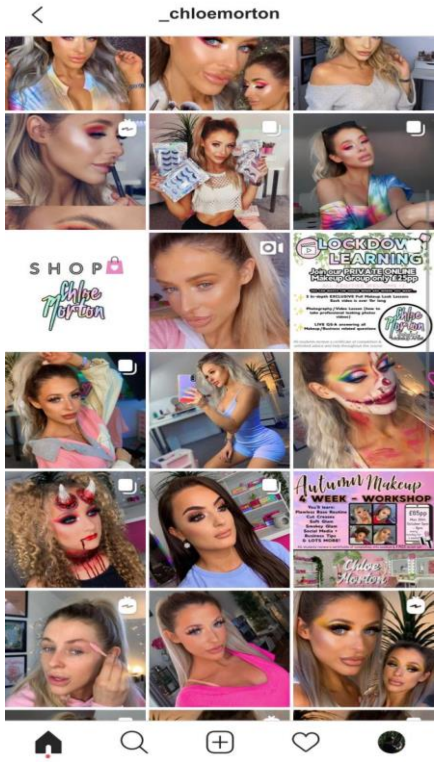
Model	Convolutional layers	Hidden layers	Optimizer	Early stop	Accuracy	Loss
VGG 16	- Filters=[30] - Activation=tanh - Padding=same - MaxPooling2D	- Dense=[125,25] - Activation=tanh - Dropout = 0.25	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	5 epochs	- Training = 93.84% - Validation = 82.35%	- Training= 0.18 - Validation=0.53
VGG 19	- Filters=[30] - Activation=tanh - Padding=valid - MaxPooling2D	- Dense=[250,125,25] - Activation=tanh - Dropout = 0.25	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	3 epochs	- Training = 84.83% - Validation = 80.25%	- Training= 0.38 - Validation=0.54
MobileNetV2	- Filters=[30] - Activation=tanh - Padding=valid - MaxPooling2D	- Dense=[250,125,25] - Activation=tanh - Dropout = 0.25	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	2 epochs	- Training = 87.94% - Validation = 83.82%	- Training= 0.45 - Validation=0.63
EfficientNetB2		- Dense=[704,352,125 25] - Activation=tanh - Dropout = 0.25	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	3 epochs	- Training = 39.24% - Validation = 43.82%	- Training = 1.33 - Validation = 1.30

CONCLUSION:

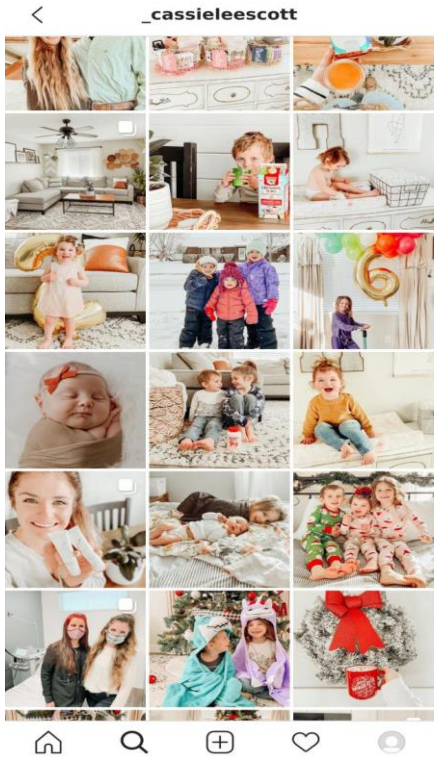
- Several models with multiple experiment were tried and played with to get the best model that goes hand in hand with the dataset.
- The best fitting model with dataset is MobileNetV2

Model	Convolutional layers	Hidden layers	Optimizer	Early stop	Accuracy	Loss
MobileNetV2	- Filters=[30] - Activation=tanh - Padding=valid - MaxPooling2D	- Dense=[250,125,25] - Activation=tanh - Dropout = 0.25	- Adam (lr=0.001) - Loss = categorical crossentropy - Metrics = accuracy	2 epochs	- Training = 87.5% - Test = 81.96%	- Training = 0.47 - Test = 0.66

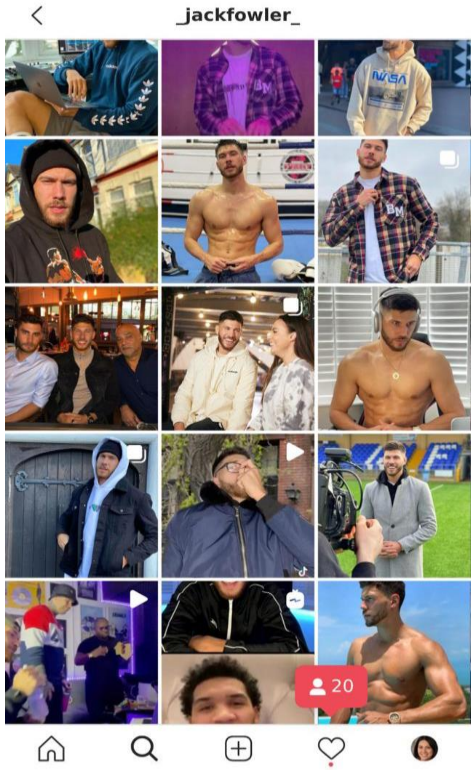
TEST SOME IMAGES



Class = Beauty
Model = Beauty



Class = Family
Model = Family



Class = Fitness
Model = Fitness

CHALLENGES

- Running time take hours to finish
- Kernal dead every couple days

THANK YOU

DO YOU HAVE ANY QUESTIONS?

By Mashael Alfehaid