# Advertising Products Based on Gender at POS

Manjunatha Setty Jayam

School of Professional Studies, Clark University

MSDA3050: Applied Machine Learning

Mr. Sanchez

October 28, 2020

**Abstract**

Early on, humans learn to distinguish one shape from another in such a way that it feels like we were born with this ability. When we try to mimic this human capability using our 0-1 marvel (machines), only then we realize the complexity of this and are left with nothing but awe for the human brain. Machine learning in layman terms is machines imitating and adapting human like behavior. The human like behavior that is expected of the machine is to 'learn from experience'. Image classification has become one of the key pilot use-cases for demonstrating machine learning. It involves the extraction of information from an image and then associating the extracted information to one or more class labels. Image classification within the machine learning domain can be approached as a supervised learning task. Here, in this project we aim to use images of humans and classify them based on gender (male or female). This classification is then further fed to the system which will then choose product adverts targeting the end consumer. This ensures that the target consumer is made aware of the product which in turn will increase the sales bringing in revenue.

*Keywords*: Machine Learning, Supervised Learning, Image Classification

## Introduction

Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. Algorithms make movie recommendations, suggest products to buy, and are increasingly used for decision making in loan applications, hiring, and dating. There are clear benefits to algorithmic decision-making; unlike people, machines can persistently consider orders of magnitude more factors than people can. One of many such areas is the domain of Computer Vision. The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc.

## Problem

To increase the sales revenue, we need to devise a smart marketing strategy. Our aim is to advertise our products in such a way that the target consumer is made aware of our products. At the sales counter, we predict the gender of the person and display relevant products on the screen behind the counter.

## Plausible Solution

The task at hand can be best handled by Convolutional Neural Network.

### *CNN (Convolutional Neural Network)*

1. CNN is a feed-forward neural network that finds a lot of its applications in the image recognition and objects recognition sectors.

2. CNN is normally constructed with 4 layers as the convolution layer, activation layer, pooling layer, and fully connected layer. The major task carried out by each layer is to extract featured and to find out patterns in the input image.

CNN works by breaking an image down into smaller groups of pixels called a filter. Each filter is a matrix of pixels. And the network does a series of calculations on these pixels comparing them against pixels in a specific pattern the network is looking for. In the first layer of a CNN, it can detect high-level patterns like rough edges and curves. As the network begins to perform more convolutions it can begin to identify specific objects like faces and animals. How does CNN know what to look for? and if its prediction is accurate? This is done through a large amount of labeled training data. When the CNN starts all of the filter values are randomized. As a result, its initial prediction makes little sense. Each time the CNN makes a prediction against a labeled data, it uses an error function to compare how close its prediction was to the image's actual label. Based on this error also known as loss function CNN updates its filter values and starts the process again. Ideally, each iteration performs with slightly more accuracy. CNN has a finite set of inputs and generates only the finite set of outputs as predicted and based on the input.

***Activation Functions***

In a computational network, the activation function of a node/neuron defines the output of that node/neuron given an input or set of inputs. It basically maps particular inputs to particular outputs. Activation functions play important roles in determining the depth and non-linearity of deep learning models.

Basically, activation functions are biologically inspired by activity in our brains when different neurons are fired or activated by different stimuli. For example, if you smell freshly baked cake

certain neurons in our brain get activated. In the same manner, if we smell something unpleasant such as rotten eggs this will cause other neurons in our brain to get activated. So, within our brain, either certain neurons are activated or not. Let us assume 1 if a certain neuron gets activated and 0 if the neuron is not activated.

sigmoid function(x) = 1/ (1+e pow (-x)). So, if we want a function that maps any input between 1 and 0 then sigmoid function is ideal one.

With a sigmoid activation function in a network, we see that output is between 0 and 1, so if the output is closer to 1 then we can say that neuron is more activated and if the output is closer to 0 then the neurons are less activated.

It is not always that the output of the activation function is between 0 and 1.

***ReLU (Rectified Linear unit):*** This is a famous and most widely used activation function which is popularly used in deep learning and neural networks. In the ReLU activation function, it transforms the input to the maximum of either 0 or the input itself.

f(x) = 0 if x <0

f(x) = x if x>=0

***Tools and libraries:***

OpenCV-Python: library of Python bindings designed to process videos and images.

NumPy: Python package that is the core library for scientific computing.

TensorFlow: AI framework for machine learning.

Keras: open-source neural-network library written in Python

## Data

The data that we need for the project is facial images of people where we can clearly identify the distinguishing features.

## Data Extraction

Google images is the best source to get a varied set of facial images data.

## Methodology

1. Obtain the data from Google Images and store it on the physical disk.
2. Data Cleaning / Data Preprocessing:
    a. The images downloaded from Google can be of different resolutions, different scale (RGB / greyscale) and so they need to be processed. This step ensures that a uniform dataset is passed on to the model for training. Having consistent data format also affects the model efficiency.
    b. Also, the data can be increased by augmenting the images. That is, we change (zoom in / zoom out / rotate / blur, etc.) the images slightly to get variations.
3. Algorithm:
    a. Here we are feeding the facial images to the convolutional neural network/model. The first convolutional layer is 2D and will have 32 filters, that will convolute the images of size (3,3). After that we are passing the inputs to ReLU activation layer that is a nonlinear layer. After this we are performing Batch Normalization. Batch normalization helps to normalize the values, some data points have high values, and some might have very less values, if we don't normalize, the high values

overshadow the low values. So, batch normalization is important for our scenario. In general batch is not necessary for all models, it mostly depends on scenario.

b. After that we apply max pooling to current network. Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map. Max pooling helps in reducing the noise in our dataset. Suppose if we have unwanted dots in our image input after the image is convoluted, the max pooling looks for the featured that are important to network and removes the other noise. Suppose if you consider the image of a man in which beard and mustache are important features and pimples are unwanted noise. After that we are adding dropout to the network. If we don't add dropout which leads to overfitting the model. And if we don't use dropout in our network, then the network will understand the training data set very well, but it fails to understand the data that is outside of the data set. So, to avoid this we are adding dropout to our network. Dropout (0.25) means 25 % of neurons will get deactivate during every front and back propagation. The model repeats the same for a different range of filters.

c. After this we are flattening the model layer. Till now Max pooling and convolution layers are 2Dimensional, to feed those layer outputs to our dense layer, those values should be in a single dimension. Now this will be feed to the output layer in which it contains classes and Sigmoid function. Whenever we come to the output layer,

we either use sigmoid or SoftMax for classifications. Because both are probability-based functions, and they give values between 0 and 1 (including).

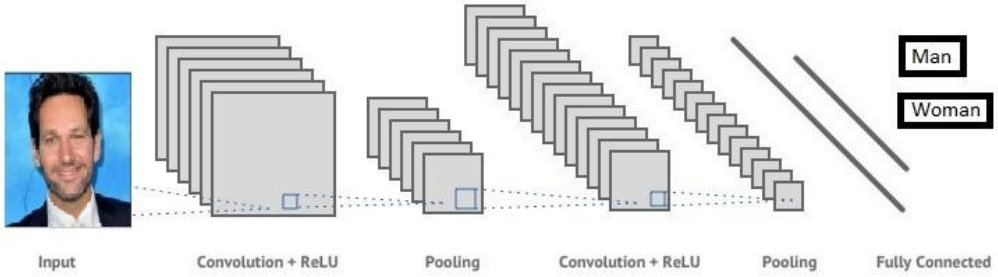


Image 1. CNN

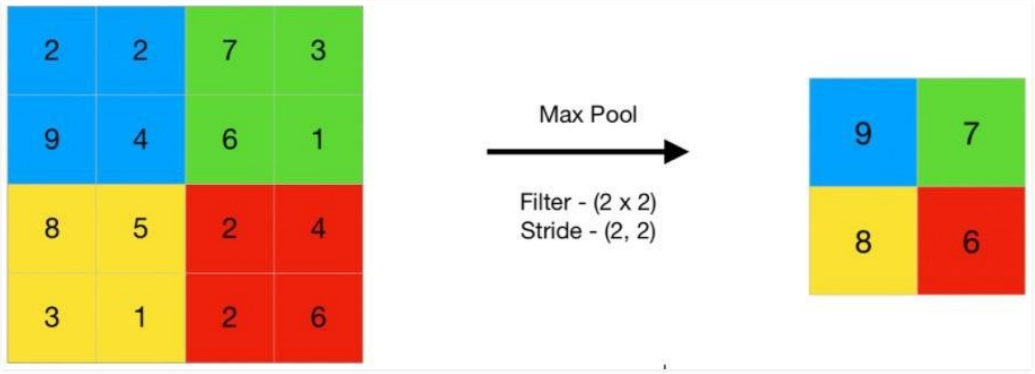


Image 2. Max Pool

d. The flowchart:

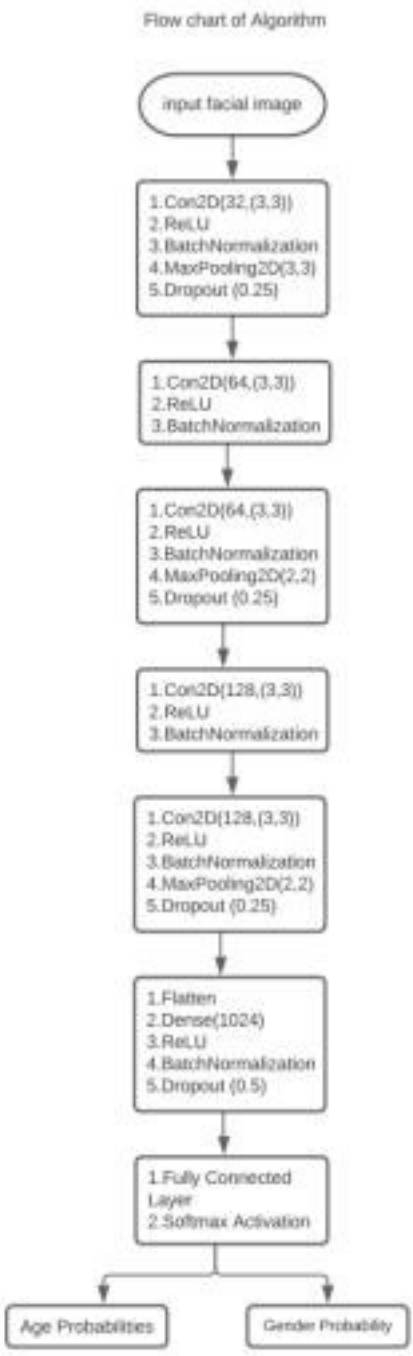


Image 3. Flowchart of Algorithm

4. Training and Testing: The dataset is divided into two subsets mainly training and testing.
    a. Ideally 80% of the data is allocated for training and the remaining 20% for testing.
    b. The model (algorithm) is trained using the training and testing (validation) dataset by iteratively running it for the given epochs. The number epochs needed to train the model is decided based on two factors: accuracy and loss. The number of epochs where the accuracy is maximum (~99% and more) and the loss is minimum (~1% or less) is finalized.
    c. The model once trained is then checked against the testing dataset. If the prediction of the algorithm is equivalent to the label of the test image, the algorithm predicted the gender correctly. Otherwise, it is considered as an error. The aim of this tests is to compare the accuracy of the model through different factors which have been iterated before and then find the most reliable model.
5. Performance of the model: Evaluation metrics help us in truly judging the performance of the model.
    a. Confusion Matrix: Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.
        i. Example: We have some sample images belonging to gender: Male / Female and we have model which predicts a gender for a given input image. On testing our model on 165 sample images, we get the following result.

| n = 165 | Predicted: Male | Predicted: Female |
|---|---|---|
| Actual: Male | 50 | 10 |
| Actual: Female | 5 | 100 |

Table 1. Confusion Matrix

ii. There are 4 important terms:

1. True Positives: The cases in which we predicted Female and the actual output was also Female.
2. True Negatives: The cases in which we predicted Male and the actual output was Male.
3. False Positives: The cases in which we predicted Female and the actual output was Male.
4. False Negatives: The cases in which we predicted Male and the actual output was Female.

iii. Accuracy for the matrix can be calculated by taking average of the values lying across the "main diagonal" i.e.

1. *Accuracy=(TruePostive + TrueNegative) / TotalSample*

Accuracy = (100 + 50) / 165 = 0.91

## Conclusion: Challenges in implementation

Challenges we might see while building and training the model are:

1. Gathering enough data for training and filtering out the quality data since the output is purely dependent on the quality of the input data.
2. Overfitting is another well-known challenge, which we should try to avoid the model in reaching that state.
3. Clutter and noise in the input data will affect the output data.
4. The output depends on the orientation, lighting conditions of the main features that are fed to the model.

## References

Christian Szegedy1, Wei Liu2, Yangqing Jia1, Pierre Sermanet1, Scott Reed3, Dragomir Anguelov1, Dumitru Erhan1, Vincent Vanhoucke1, Andrew Rabinovich4 (2015). Going Deeper with Convolutions. https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_20 15_CVPR_paper.pdf

Lowe D.G. (2004, November) Distinctive Image Features from Scale-Invariant Keypoints. https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94

Mayibongwe H. BayanaSerestina ViririEmail authorRaphael Angulu. (2018, November). Gender Classification Based on Facial Shape and Texture Features. https://link.springer.com/chapter/10.1007/978-3-030-03801-4_15

Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S., & Babu, R. V. (2016). A Taxonomy of Deep Convolutional Neural Nets for Computer Vision. https://www.frontiersin.org/articles/10.3389/frobt.2015.00036/full

Wang, Y., & Wu, Y. Scene Classification with Deep Convolutional Neural Networks. https://www.semanticscholar.org/paper/Scene-Classification-with-Deep-Convolutional-Neural-Wang-Wu/05b3d0ece3c05be45b1ae7db3b43123befae3474?p2df

Yang, J., Jiang, Y. G., Hauptmann, A. G., & Ngo, C. W. (2007, September). Evaluating bag-of-visual-words representations in scene classification. https://dl.acm.org/doi/abs/10.1145/1290082.1290111

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2014) "Object detectors emerge in Deep Scene CNNs". https://dspace.mit.edu/handle/1721.1/96942