Team 10- A.V.Ramesh, S.Waghmare, L.Sridevi

Department of Electrical & Computer Engineering
University of Illinois at Chicago

# Head Pose Estimation

# Outline

# Problem at hand - Estimating head position


## Problem at hand

Determine the head position(Up/Down/Center and Left/Right/Center) given a face image. Why is this problem important?

- Dataset : Prima head pose
- Model : Convolutional Neural Network
- Languages : Python 3
- Tools : TensorFlow
- Hardware : Google cloud GPU

# Outline

# Typical approach

- How does a human estimate head pose?- We learn this skill right from birth.
- Computers see only numbers from the image !!! We need to help the model learn/understand!!
- This can be achieved by learning features - Design a filter and convolve with the image to extract its feature.

Visualization of the receptive field

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 30 |
| 0 | 0 | 0 | 0 | 50 | 50 | 50 |
| 0 | 0 | 0 | 20 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |
| 0 | 0 | 0 | 50 | 50 | 0 | 0 |

Pixel representation of the receptive field

$*$

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 30 | 0 |
| 0 | 0 | 0 | 0 | 30 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 30 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

## Challenges

Who gives the filter?? Thanks to Neural Networks!

# Outline

# Convolutional Neural Network Model(CNN)

CNN model learns the filter(weight values) from the dataset and convolves the filter with the input image to estimate the output.

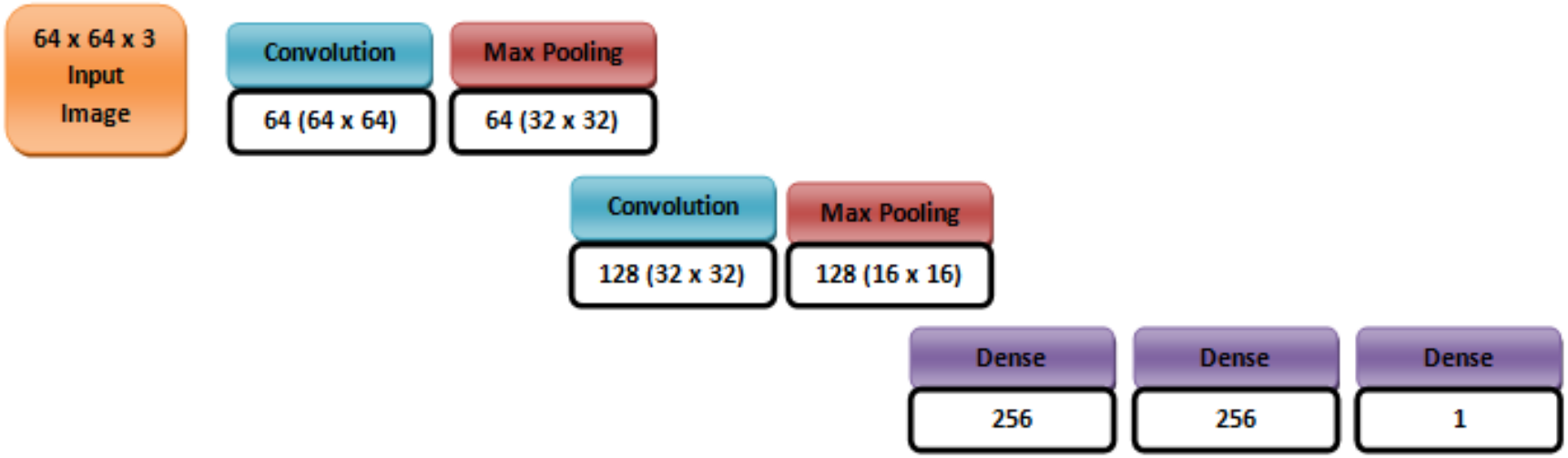
## Steps - CNN weights learning

Initializes weights with random values

- **Forward pass** - Runs the model with a given input(image) and makes a prediction(output)
- **Loss function** - Computes the error between the true output and predicted output.
- **Backward pass** - Propagates the loss on the network to determine which weights contributed to the loss.
- **Weight update**- Updates the weight values.

Repeats the steps until the loss is minimized.

# Model Architecture




**CNN Architecture**

# Outline

# Dataset

## Dataset Description

- 2790 face images of 15 subjects- 2 series of 93 images of discrete poses for each person
- Variations in pan and tilt (pitch and yaw) angles in the range [-90,90].
- Images in the dataset have occlusions such as glasses and variations such as skin color.
- Background is willingly neutral and uncluttered in order to focus on face operations

## Challenge

Obtaining the Head Pose dataset with head angle(tilt and pan) is very hard!!

# Outline

# Our Model

- Typically this is solved as a regression problem to achieve accuracy. But obtaining dataset is very hard.
- Therefore we decided to pose it as a **classification problem** - like Humans do! - Classifying head poses is a trickier problem than computing angles here!

## Our approach and Challenges

- We started with 25 classes for a much precise classification - We got very low accuracy - we figured out that the reason was data deficit(every class had only 100 examples)
- We reduced the classes to 9 - accuracy improved, but not good enough

## Idea- Divide and Conquer

- Yes! Divide the problem into 2 independent classification problems - learn independent filters for tilt and pan.
- Now each model has only 3 classes to learn, around 900 examples per class.

# Outline

# Results

## Independent model accuracy as of today

| Tilt | Pan |
|---|---|
| 0.77 | 0.48 |

## To improve accuracy

- Parameter tuning
- Data augmentation