Hand Signals With Visual Data

CAPSTONE PROJECT

PROJECT PROPOSAL BY
AYUSHYA, OMAR, CHRIS, GERRY

HAND SIGNALS WITH VISUAL DATA

OVERVIEW

Our Idea

Our team would like to build a software that could allow for hand gestures to control a computer with the use of a camera in a way similar to the mouse.

Role of Deep Learning

Deep learning would allow for the recognition of hand gestures and number of fingers. The combination of these two feature sets would allow for a fully customizable gesture system.

HAND SIGNALS WITH VISUAL DATA

OUR IDEA

Challenges

Optimizing for on device deep learning

Training a Convolutional Neural Network to recognize gestures like scrolling, pinching, waving, and rotating

Real time finger detection allowing for counting the number of fingers being held up, this could be additionally be trained to detect a folded finger

Gathering a sufficiently large dataset to train new models

PROGRESS UPDATE

HAND SIGNALS WITH VISUAL DATA
PROGRESS UPDATE

What we have tried

- ➔ Trying and testing available libraries (OpenPose based on CMU Caffe)
- ➔ Looking for pre-trained models and datasets

Our plan moving forward

From our tests so far we are leaning towards using OpenPose and CMU's Hand Database. This would allow us with a sufficiently large and varied database allowing for the customization that we plan on delivering.

TECHNOLOGIES USED

TECHNOLOGIES USED

Electron Framework
Framework for creating native applications with web technologies

Node.js Runtime
An asynchronous event driven JavaScript runtime for application backends

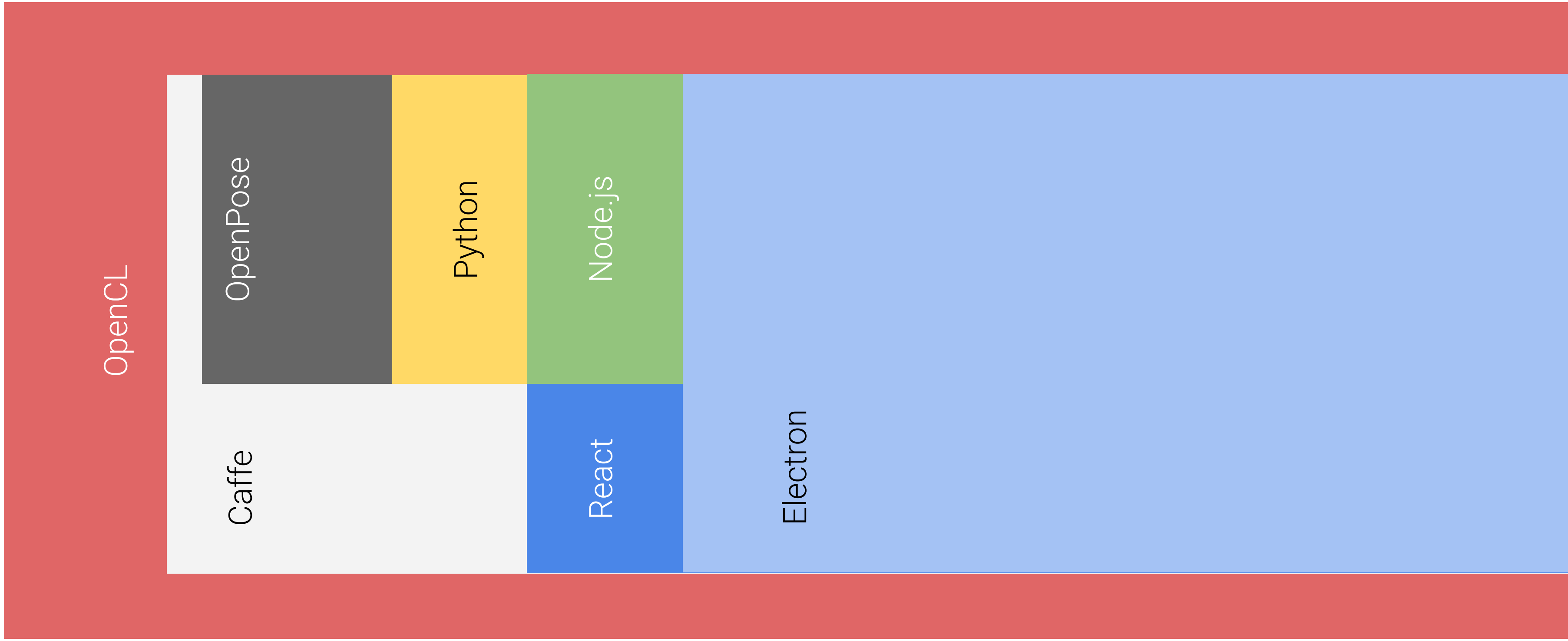
React
React is a JavaScript library for building user interfaces

OpenPose
Real-time system to jointly detect human body, hand, facial, and foot keypoints

OpenCL
A framework for programs that execute across hardware accelerators

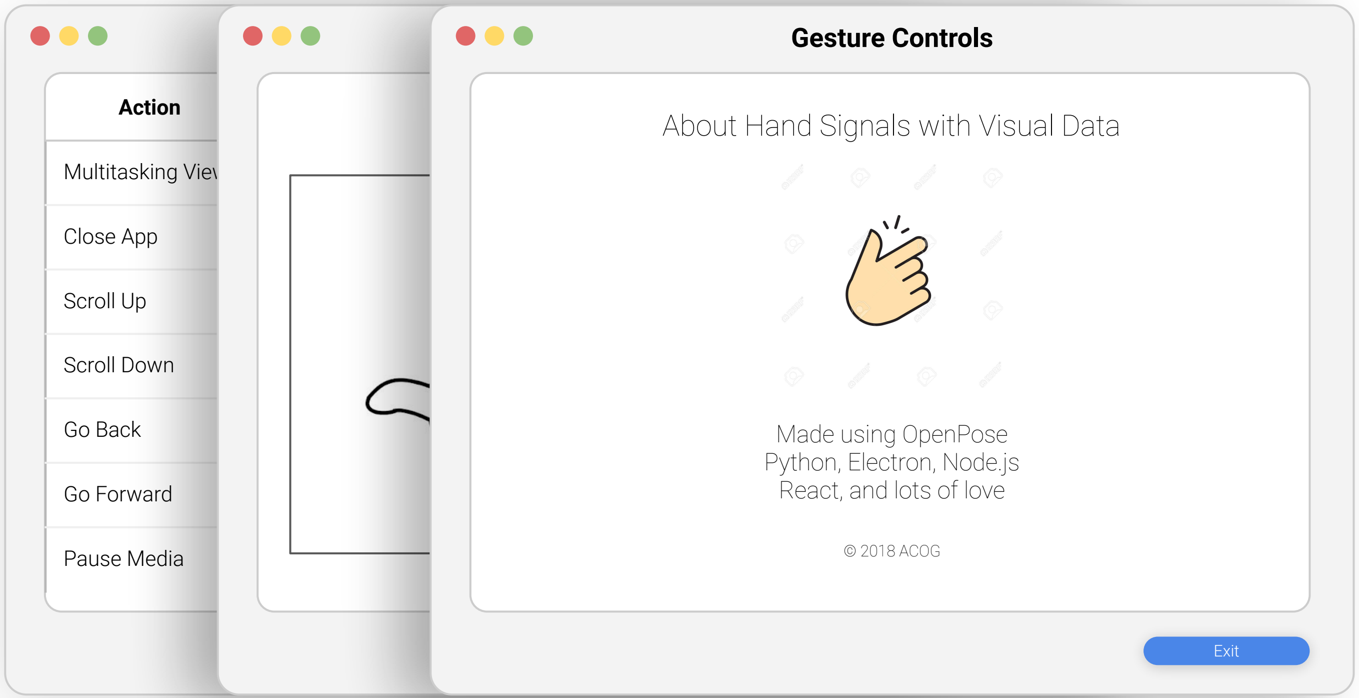
Python
Interpreted high-level language of choice for OpenPose API

TECHNOLOGIES USED
HOW IT WILL WORK



TECHNOLOGIES USED

HOW IT WILL LOOK



Action

Multitasking Viev

Close App

Scroll Up

Scroll Down

Go Back

Go Forward

Pause Media

Gesture Controls

About Hand Signals with Visual Data

Made using OpenPose
Python, Electron, Node.js
React, and lots of love

© 2018 ACOG

TECHNOLOGIES USED
SOFTWARE FEATURES

Input
The user will be able to use any camera input - this will make the program accessible on every platform.

Target Platforms
The software should work on all major desktop operating systems - Windows, MacOS, and Linux.

Customization
The software will allow for triggering built-in OS shortcuts from custom gesture and finger count combinations.

Privacy
The software will only collect images from users if given explicit permission - the models will be stored locally on the users' machines allowing for private use.

MAKING THE TEAM
TEAM ROLES

Frontend and Design

- ➔ Use Electron Javascript framework to easily build native cross platform applications
- ➔ Design a modern responsive and easy to use interface

Data Management

- ➔ Take advantage of Python packages to collect necessary data
- ➔ Make sure data is consistent

Backend APIs

- ➔ Create OTA model update system
- ➔ Create gesture and action linking system
- ➔ Implement background services

Gesture Recognition Model Research & Development

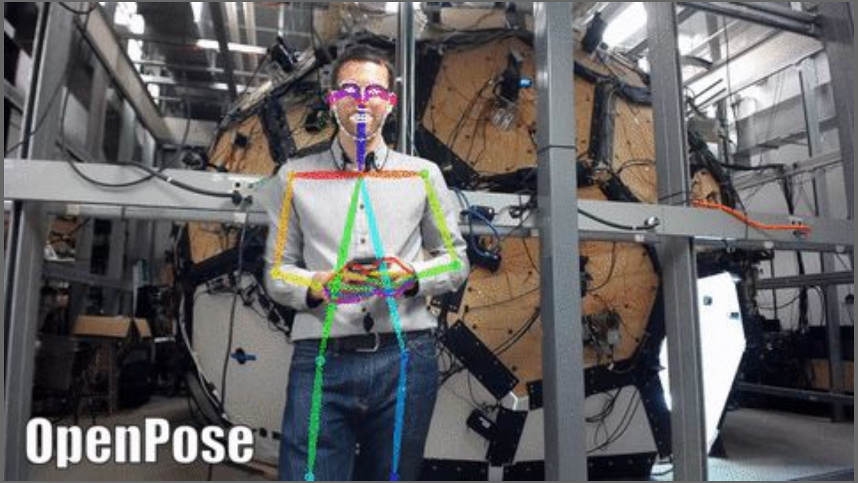
- ➔ A team effort to create models that will allow real-time gesture detection on the end user's local environment

OPENPOSE
WHAT IT IS

The first real-time multi-person system to jointly detect human body, hand, facial, and foot keypoints.

Library Built On Caffe

Caffe is a deep learning framework built by the AI division at UC Berkeley - the version used by OpenPose however, is a version modded by Carnegie Mellon University.



OPENPOSE

HOW IT WORKS

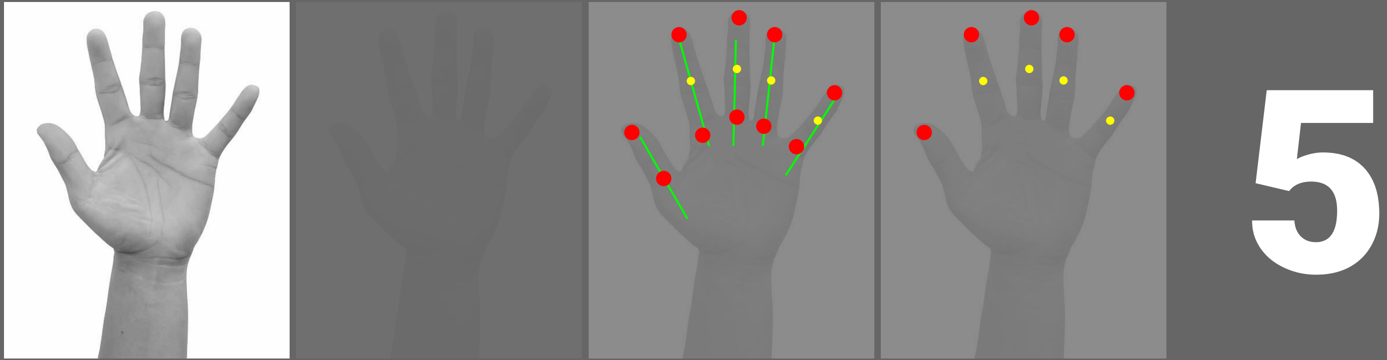
The main class for pose extraction is called `PoseExtractorCaffe()`

The inputs for pose extraction are

`netInputSize`	Input size of the CNN (multiple of 16)
`netOutputSize`	Resolution of the last layer of CNN
`outputSize`	Output size of the final rendered image
`scaleNumber`	Number of scales to process
`scaleGap`	Gap between successive scales. `[1, 1-scaleGap*scaleNumber]`
`poseModel`	Affects the number of body parts to render

OPENPOSE

HOW WE WILL USE IT



As an end goal, we will use the existing model from OpenPose and modify it to predict the number and position of fingers. We can use the hand database made available by CMU Panoptic Dataset for re-training purposes.

CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network is an algorithm which takes in an input image, assigns weights and biases to object in the image and be able to differentiate one from another.

CNN applies a series of filter to raw pixel data of an image to extract and learn higher-level features

COMPONENTS OF A CNN

Input data
CMU Panoptic hand gesture dataset

Loading data
Load images into an array for storing the image and the other array for the label

Creating CNN Model
To simplify the idea, we’re using a linear regression model to create a simple model.

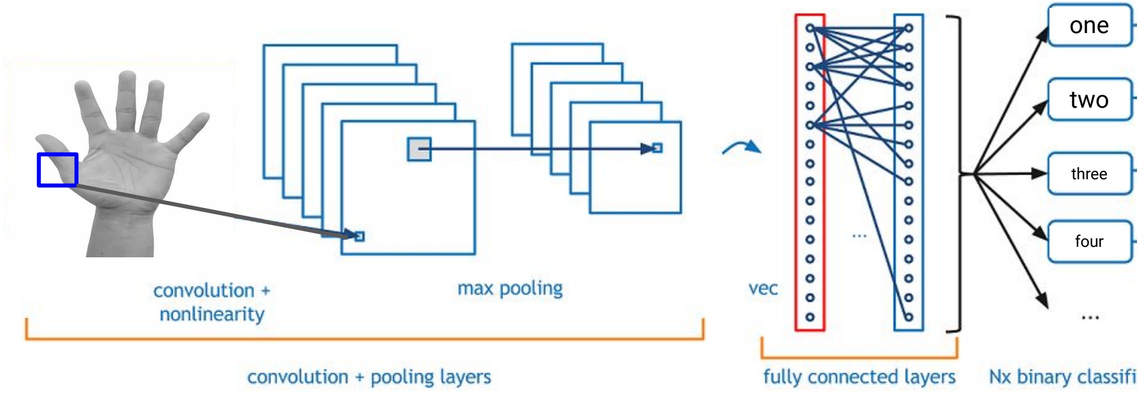
CONVOLUTIONAL NEURAL NETWORK COMPONENTS

THREE MAIN COMPONENTS

Convolutional Layer: applies a specific number of convolutional filters to the image

Pooling layer: downsample the extracted image data to reduce the dimensionality of the feature map in order to decrease processing time

Fully connected layer: Perform classification on the feature extracted by the convolutional layer and downsampled by the pooling layer.



CONVOLUTIONAL NEURAL NETWORK
TRAINING MODEL

Requirements to train the network

Loss Function

`softmax_cross_entropy_with_logits` function, which take in the output of the final layer and the training target vector and compares them

Optimizer

Takes in a learning rate that minimizes the loss function

Correct Prediction and Accuracy

Correct prediction operation returns value determining whether the neural network has correctly predicted the digit. Accuracy operation uses the result obtained from correct prediction operation and return the mean accuracy of the tensor

Perform the training accuracy

Using the dataset, run the loss function and optimizer operations for the number of batch and epoch that was initialized and calculate the accuracy obtained at each step

DEMO

REFERENCES

OpenPose

https://github.com/CMU-Perceptual-Computing-Lab/openpose

Handtrack.js

https://github.com/victordibia/handtrack.js/
Library for prototyping real-time hand detection

Gesture Recognition

https://github.com/asingh33/CNNGestureRecognizer
https://github.com/avidLearnerInProgress/hand-gesture-recognition
Projects with examples for gestures recognition

TensorFlow Examples

https://github.com/tensorflow/examples
https://github.com/tensorflow/examples/.../gesture_classification
General examples from tensorflow on gesture classification and object recognition

Kaggle Dataset

https://www.kaggle.com/.../hand-gesture-recognition-database-with-cnn/
Database is composed by 10 different hand-gestures

THANK YOU