Technische
Universität
Berlin

Real-time hand tracking

Yannick Schickel, Armin Jaber, Dilip | Project „Hot Topics in Computer Vision“: Video Understanding | Final report

Table of Contents

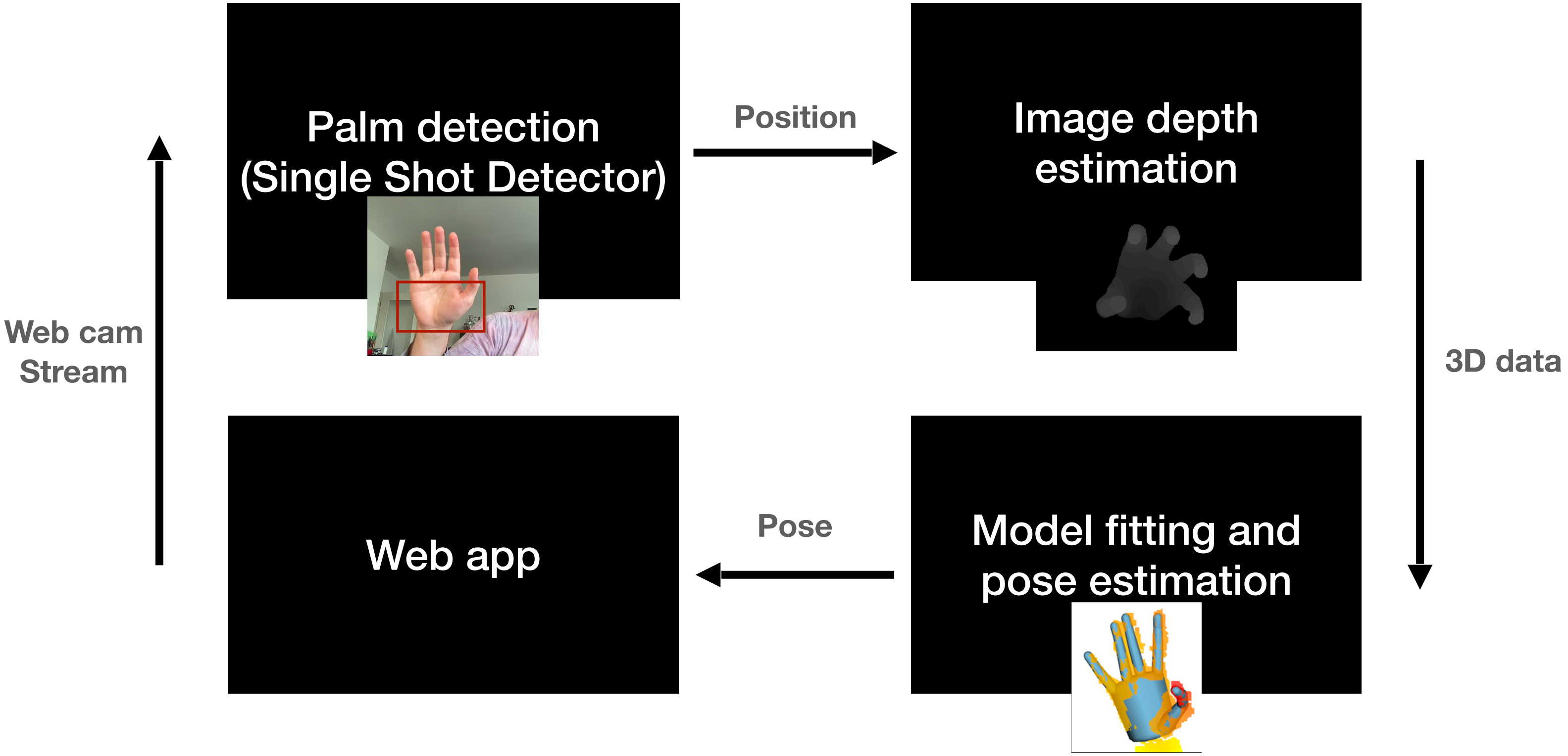
- Introduction
- Topic and Goals
- Starting conditions
- Implementation
- Achievements
- Live-Demo

Introduction

- hand articulation is crucial in human’s non-verbal communication
- Tracking hands in real-time enables stronger human-machine-interfaces
- Two approaches: **appearance based** vs. **model based**
- Two approaches: **data-driven** vs. **knowledge-based**

Goals

- Real-time hand tracking on mobile web app
- Prototype Plan:



Palm detection (Single Shot Detector)

Position

Image depth estimation

3D data

Model fitting and pose estimation

Pose

Web app

Web cam Stream

Starting conditions

Palm detection:

- Model decision from Object Detection Model Zoo

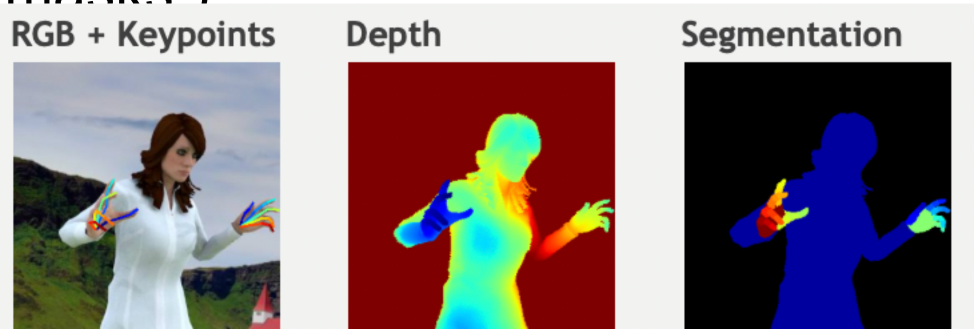
(https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md)

—> ssd_mobilenet_v2_320x320_coco17_tpu-8 (speed=19ms)

- Dataset from Uni Freiburg

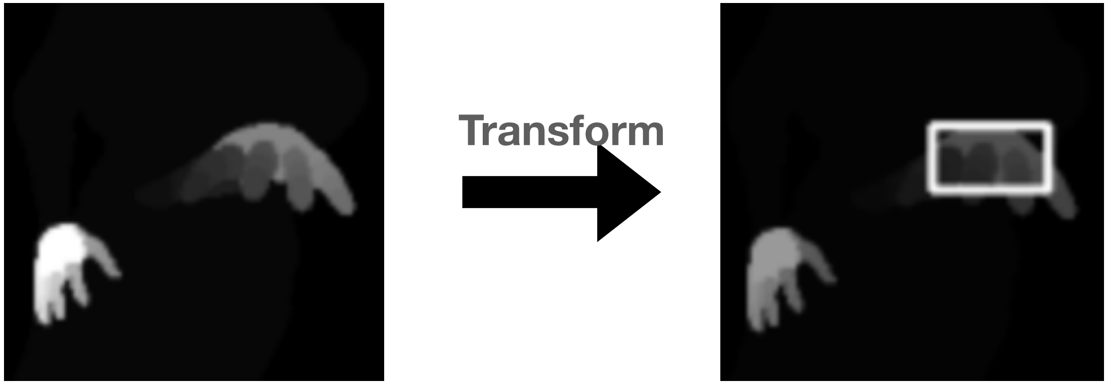
(https://lmb.informatik.uni-freiburg.de/resources/datasets/RenderedHandposeDataset.en.html)

—>Rendered Handpose Dataset
(41258 training samples with RGB, Depth map and segmentation masks)



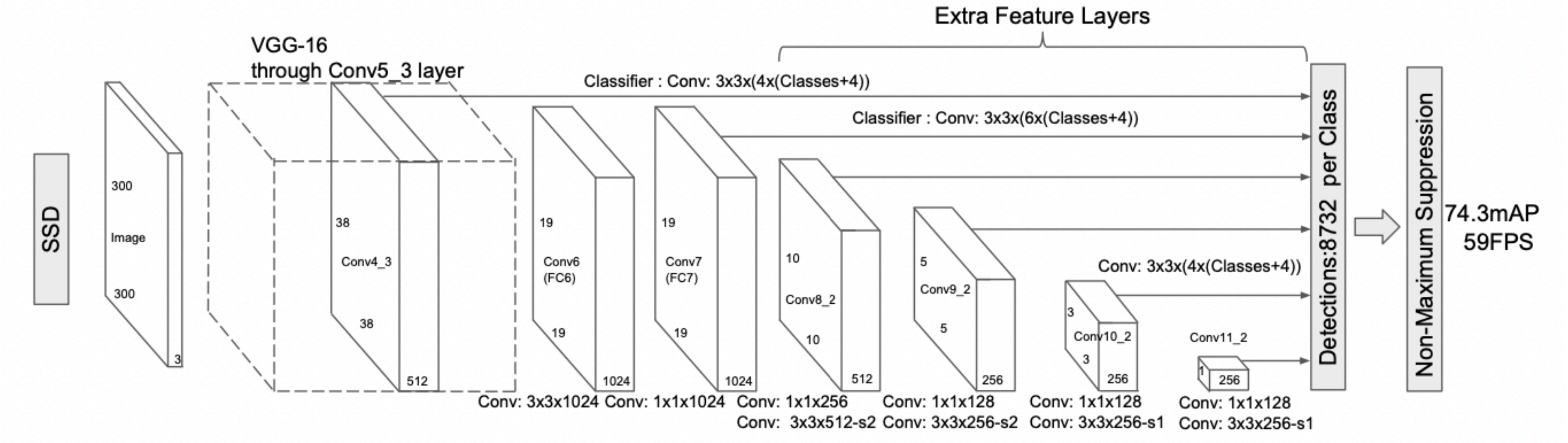
Implementation(Preprocessing)

1. Setting up environment (in Google Colab)
2. Installing Object Detection API
3. Dataset Pre-processing:
   Segmentation Mask to Boundingbox Transform



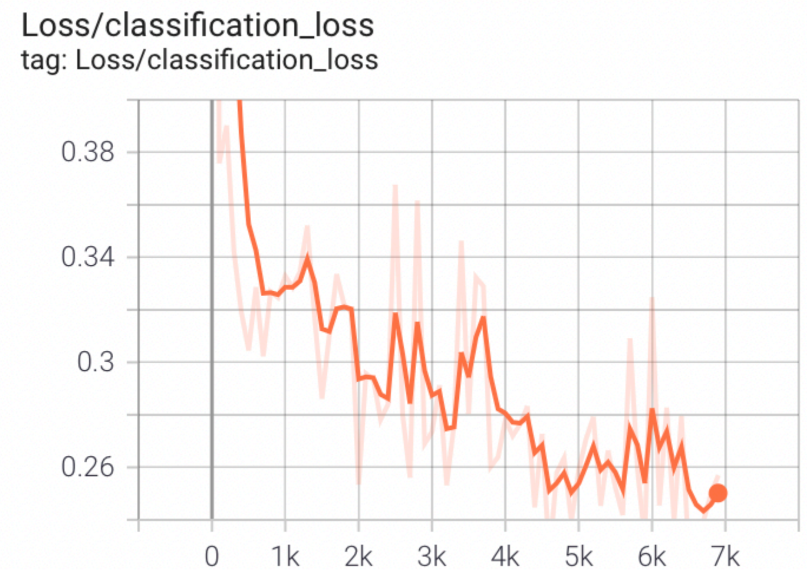
Implementation(SSD model)

- Base network (architecture for high quality image classification)
- Multi-scale feature maps for detection
- Convolutional predictors/classifiers for detection
- https://arxiv.org/pdf/1512.02325.pdf

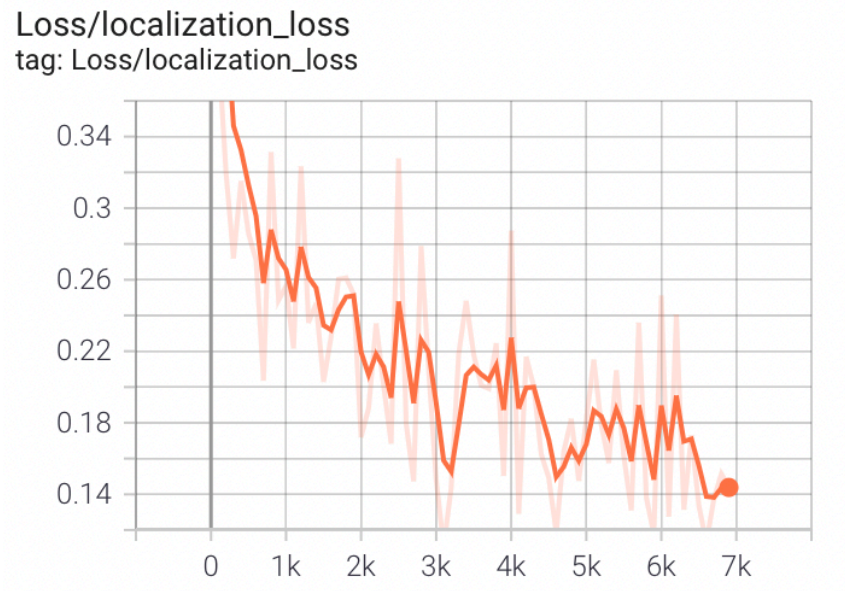


Implementation(Training the model)

- Batch size = 8, Epochs = 7000



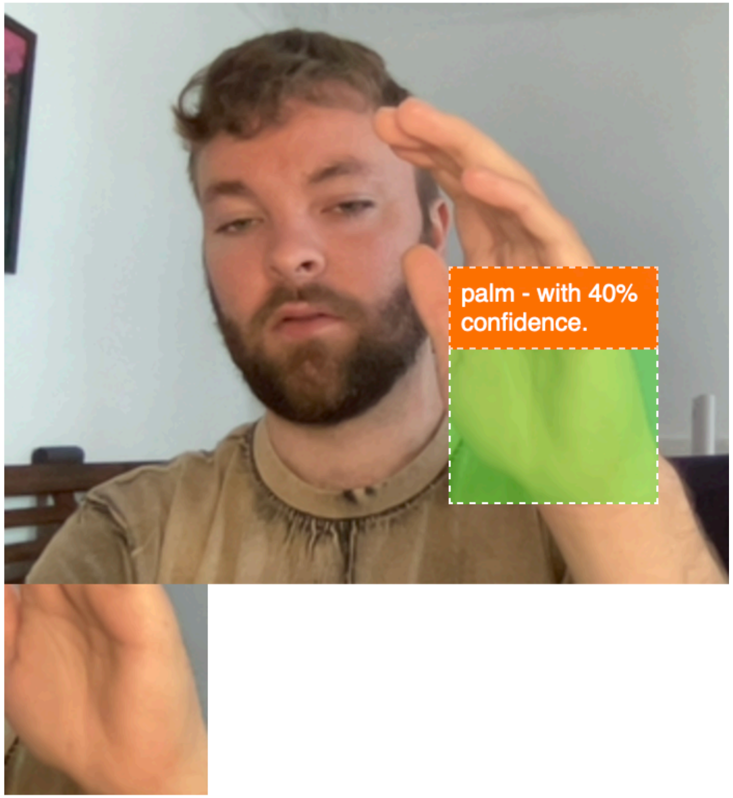
Loss/classification_loss
tag: Loss/classification_loss



Loss/localization_loss
tag: Loss/localization_loss

Implementation(Post-processing)

- Non-Maximum suppression
- Cropping Hand Region

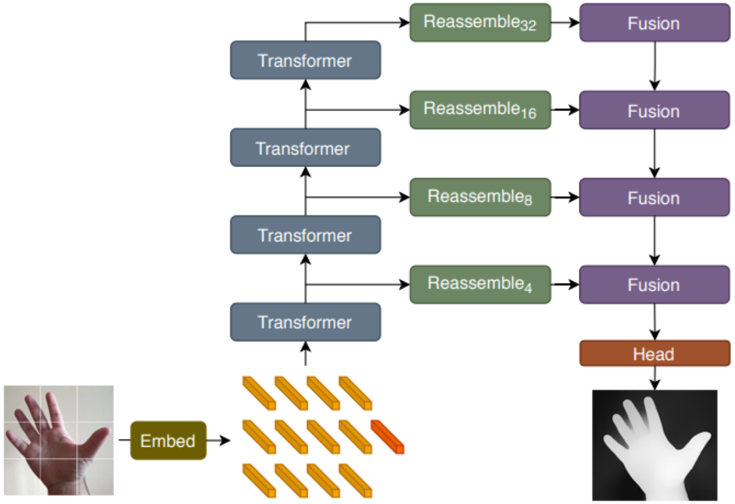


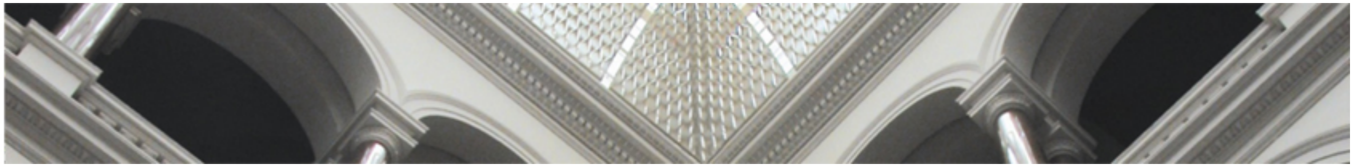
Achievements (Depth Estimation)

Vision Transformers for Dense Prediction from Intel lab.

- This Model work in three layers:

1. Transformer encoder:
   - The image is transform into tokens by extracting their feature with the feature extractor (orange token)
   - The embedded image is attached with their positional argument and add as a read only token (red token)
2. Resemble the tokens into multiple layers.
3. Fuse the layers to generate a depth image.

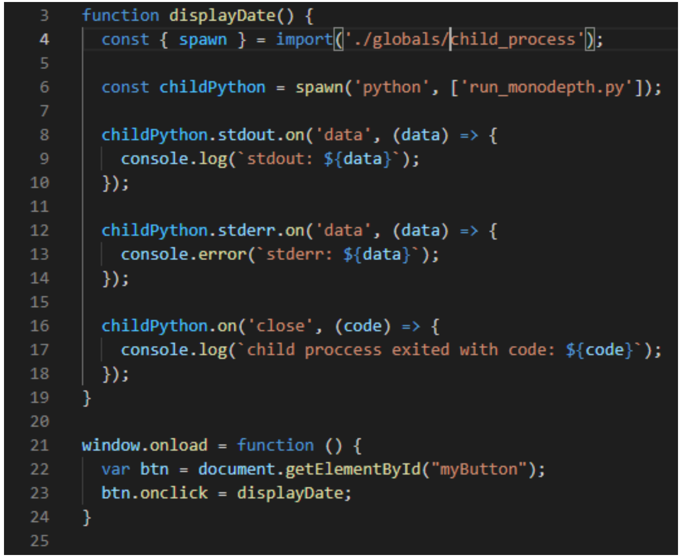




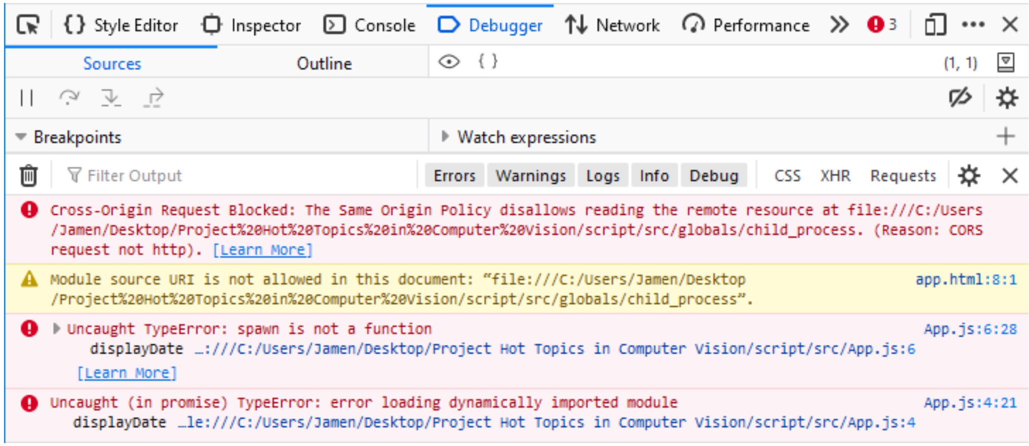
Depth Estimation (Code)

Challanges:

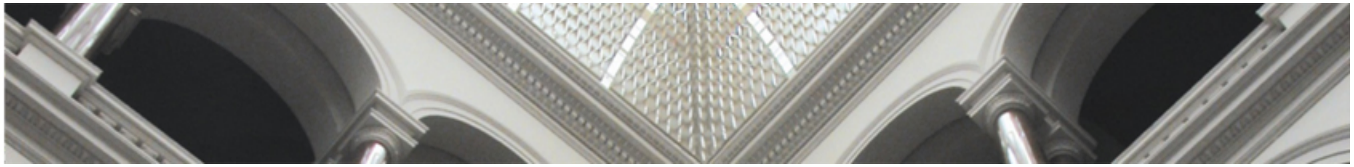
- Input and output are single images
- Not always use the GPU processor
- Incompatible for integrating the model in a web Application



```
function displayDate() {
  const { spawn } = import('./globals/child_process');

  const childPython = spawn('python', ['run_monodepth.py']);

  childPython.stdout.on('data', (data) => {
    console.log(`stdout: ${data}`);
  });

  childPython.stderr.on('data', (data) => {
    console.error(`stderr: ${data}`);
  });

  childPython.on('close', (code) => {
    console.log(`child proccess exited with code: ${code}`);
  });
}

window.onload = function () {
  var btn = document.getElementById("myButton");
  btn.onclick = displayDate;
}
```



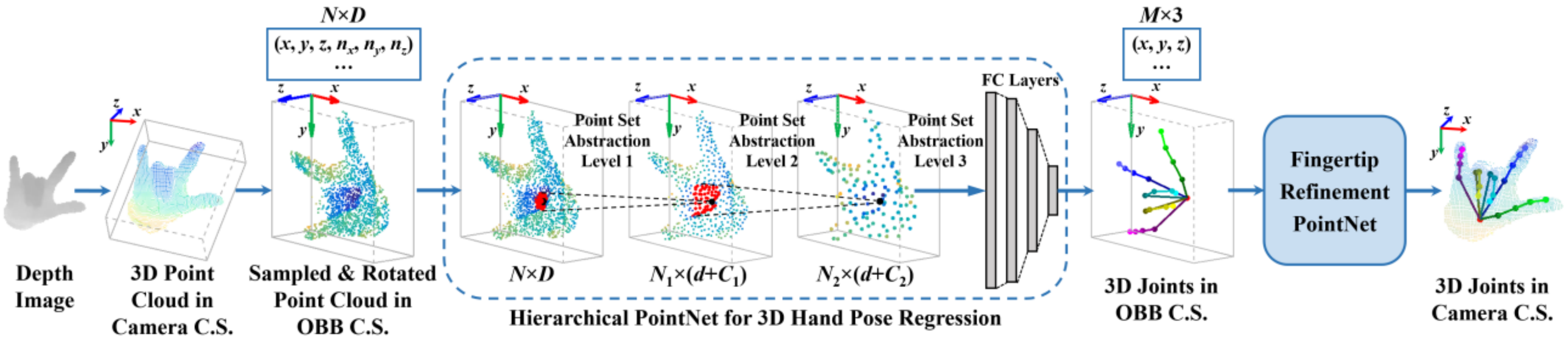
```
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at file:///C:/Users
/Jamen/Desktop/Project%20Hot%20Topics%20in%20Computer%20Vision/script/src/globals/child_process. (Reason: CORS
request not http). [Learn More]
Module source URI is not allowed in this document: “file:///C:/Users/Jamen/Desktop                  app.html:8:1
/Project%20Hot%20Topics%20in%20Computer%20Vision/script/src/globals/child_process”.
▶ Uncaught TypeError: spawn is not a function                                                     App.js:6:28
    displayDate …:///C:/Users/Jamen/Desktop/Project Hot Topics in Computer Vision/script/src/App.js:6
   [Learn More]
Uncaught (in promise) TypeError: error loading dynamically imported module                          App.js:4:21
   displayDate …le:///C:/Users/Jamen/Desktop/Project Hot Topics in Computer Vision/script/src/App.js:4
```



Hand Fitting Model

Hand PointNet model:

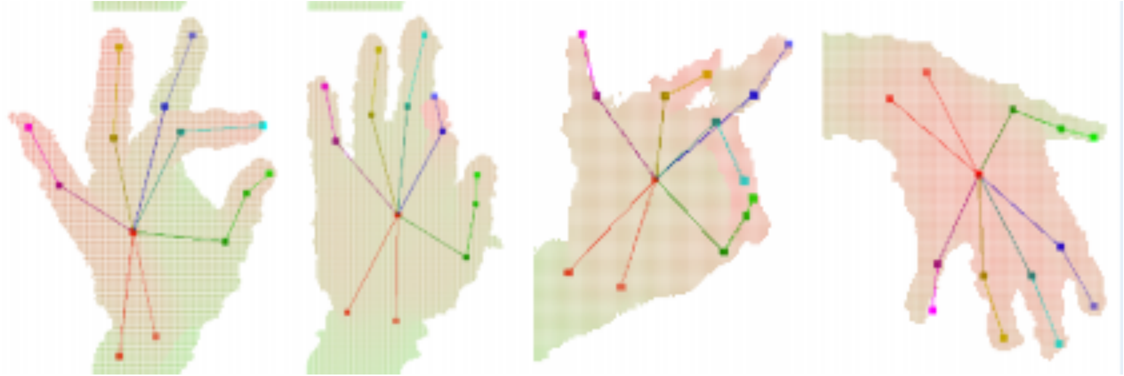
- depth image or a 3d volume as an input
- Generating a 3d dimensional point cloud from the depth image
- Create a bunding box
- Input 3d dimetional point into the hierarchical PointNet
- Input estimated finger tips point into basic PointNet


N×D
(x, y, z, n_x, n_y, n_z)
...
M×3
(x, y, z)
...
z
x
y
Point Set Abstraction Level 1
Point Set Abstraction Level 2
Point Set Abstraction Level 3
FC Layers
Fingertip Refinement PointNet
Depth Image
3D Point Cloud in Camera C.S.
Sampled & Rotated Point Cloud in OBB C.S.
N×D
N1×(d+C1)
N2×(d+C2)
Hierarchical PointNet for 3D Hand Pose Regression
3D Joints in OBB C.S.
3D Joints in Camera C.S.

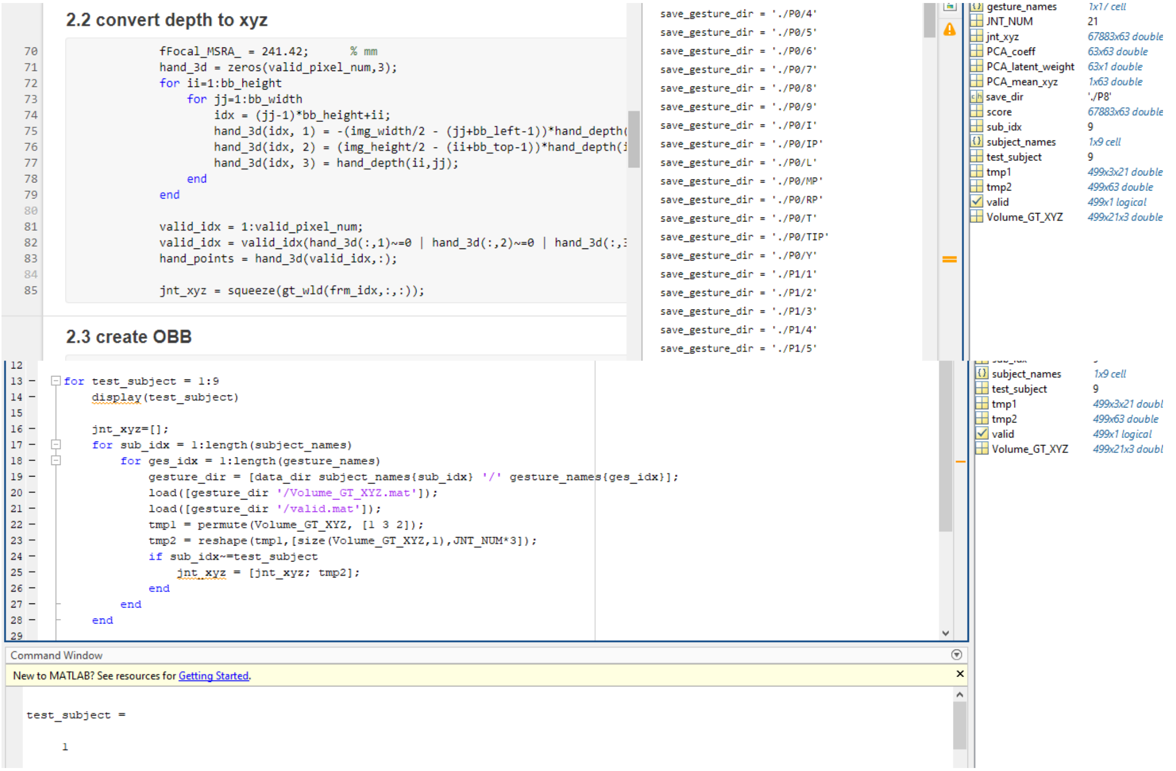
Hand Fitting Model

Hand Pose Estamition

- Using the deep neural Networks-based discriminative approach
  - Joints constrains
  - Different angle view
  - Perspective



Hand Fitting Model (Code sample)



2.2 convert depth to xyz

```
fFocal_MSRA_ = 241.42;      % mm
hand_3d = zeros(valid_pixel_num,3);
for ii=1:bb_height
    for jj=1:bb_width
        idx = (jj-1)*bb_height+ii;
        hand_3d(idx, 1) = -(img_width/2 - (jj+bb_left-1))*hand_depth(
        hand_3d(idx, 2) = (img_height/2 - (ii+bb_top-1))*hand_depth(i
        hand_3d(idx, 3) = hand_depth(ii,jj);
    end
end

valid_idx = 1:valid_pixel_num;
valid_idx = valid_idx(hand_3d(:,1)~=0 | hand_3d(:,2)~=0 | hand_3d(:,3

hand_points = hand_3d(valid_idx,:);

jnt_xyz = squeeze(gt_wld(frm_idx,:,:));
```

2.3 create OBB

```
save_gesture_dir = './P0/4'
save_gesture_dir = './P0/5'
save_gesture_dir = './P0/6'
save_gesture_dir = './P0/7'
save_gesture_dir = './P0/8'
save_gesture_dir = './P0/9'
save_gesture_dir = './P0/I'
save_gesture_dir = './P0/IP'
save_gesture_dir = './P0/L'
save_gesture_dir = './P0/MP'
save_gesture_dir = './P0/RP'
save_gesture_dir = './P0/T'
save_gesture_dir = './P0/TIP'
save_gesture_dir = './P0/Y'
save_gesture_dir = './P1/1'
save_gesture_dir = './P1/2'
save_gesture_dir = './P1/3'
save_gesture_dir = './P1/4'
save_gesture_dir = './P1/5'
```

gesture_names	1x17 cell
JNT_NUM	21
jnt_xyz	67883x63 double
PCA_coeff	63x63 double
PCA_latent_weight	63x1 double
PCA_mean_xyz	1x63 double
save_dir	'./P8'
score	67883x63 double
sub_idx	9
subject_names	1x9 cell
test_subject	9
tmp1	499x3x21 double
tmp2	499x63 double
valid	499x1 logical
Volume_GT_XYZ	499x21x3 double

```

for test_subject = 1:9
    display(test_subject)

    jnt_xyz=[];
    for sub_idx = 1:length(subject_names)
        for ges_idx = 1:length(gesture_names)
            gesture_dir = [data_dir subject_names{sub_idx} '/' gesture_names{ges_idx}];
            load([gesture_dir '/Volume_GT_XYZ.mat']);
            load([gesture_dir '/valid.mat']);
            tmp1 = permute(Volume_GT_XYZ, [1 3 2]);
            tmp2 = reshape(tmp1,[size(Volume_GT_XYZ,1),JNT_NUM*3]);
            if sub_idx~=test_subject
                jnt_xyz = [jnt_xyz; tmp2];
            end
        end
    end

```

subject_names	1x9 cell
test_subject	9
tmp1	499x3x21 doubl
tmp2	499x63 double
valid	499x1 logical
Volume_GT_XYZ	499x21x3 doubl

Command Window

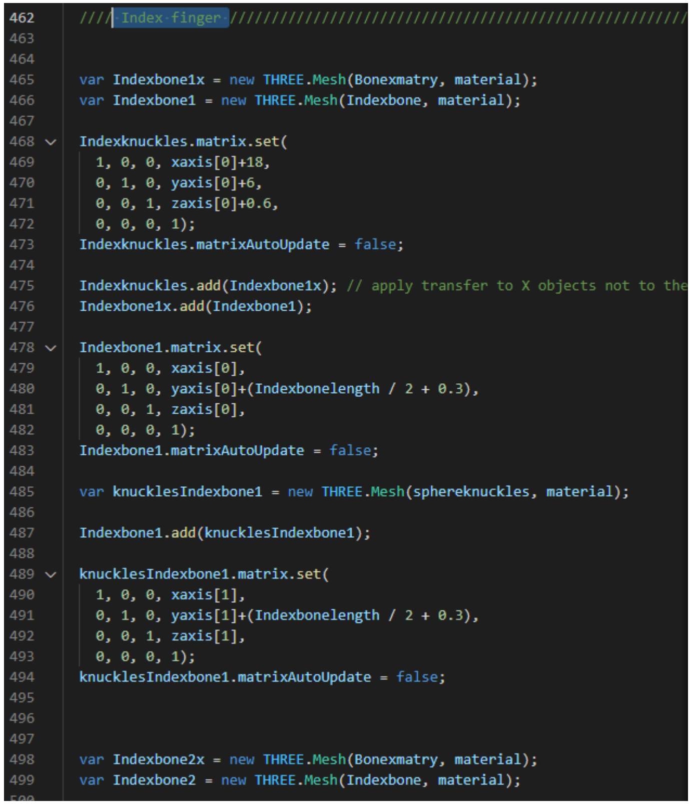
New to MATLAB? See resources for Getting Started.

```
test_subject =

     1
```

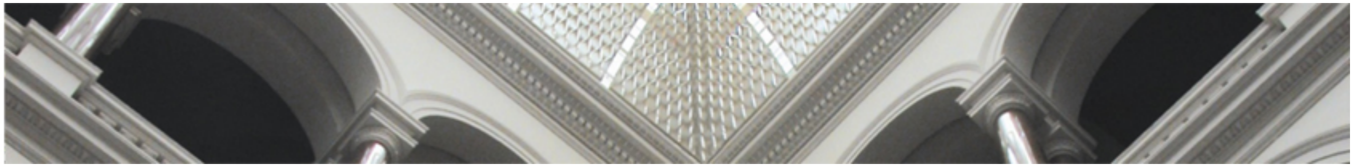
3d Hand Model

Creating a 3d Model using threejs Library

- By adding Geometers together and put them in their respektive Position on the Palm

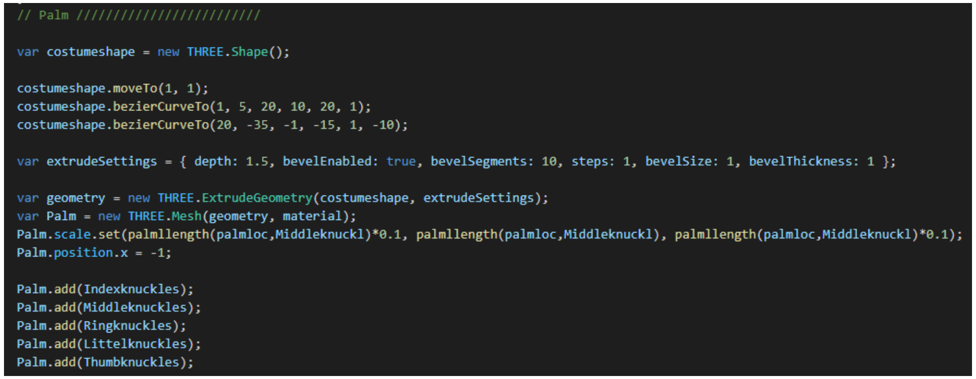


```
//// Index finger //////////////////////////////////////////////////////


var Indexbone1x = new THREE.Mesh(Bonexmatry, material);
var Indexbone1 = new THREE.Mesh(Indexbone, material);

Indexknuckles.matrix.set(
  1, 0, 0, xaxis[0]+18,
  0, 1, 0, yaxis[0]+6,
  0, 0, 1, zaxis[0]+0.6,
  0, 0, 0, 1);
Indexknuckles.matrixAutoUpdate = false;

Indexknuckles.add(Indexbone1x); // apply transfer to X objects not to the
Indexbone1x.add(Indexbone1);

Indexbone1.matrix.set(
  1, 0, 0, xaxis[0],
  0, 1, 0, yaxis[0]+(Indexbonelength / 2 + 0.3),
  0, 0, 1, zaxis[0],
  0, 0, 0, 1);
Indexbone1.matrixAutoUpdate = false;

var knucklesIndexbone1 = new THREE.Mesh(sphereknuckles, material);

Indexbone1.add(knucklesIndexbone1);

knucklesIndexbone1.matrix.set(
  1, 0, 0, xaxis[1],
  0, 1, 0, yaxis[1]+(Indexbonelength / 2 + 0.3),
  0, 0, 1, zaxis[1],
  0, 0, 0, 1);
knucklesIndexbone1.matrixAutoUpdate = false;



var Indexbone2x = new THREE.Mesh(Bonexmatry, material);
var Indexbone2 = new THREE.Mesh(Indexbone, material);
```

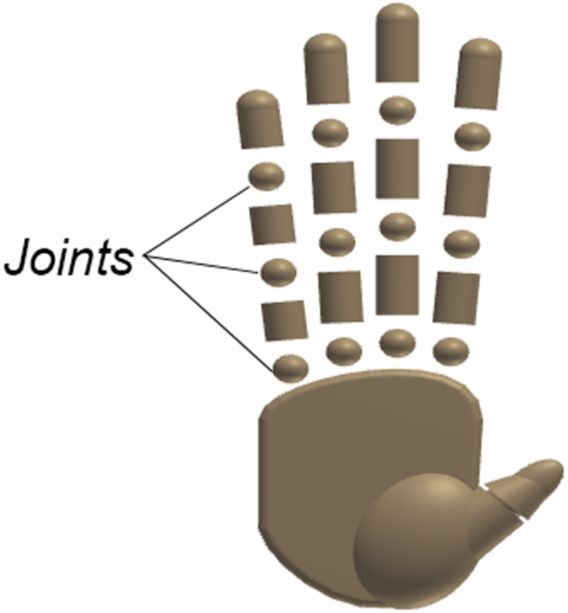


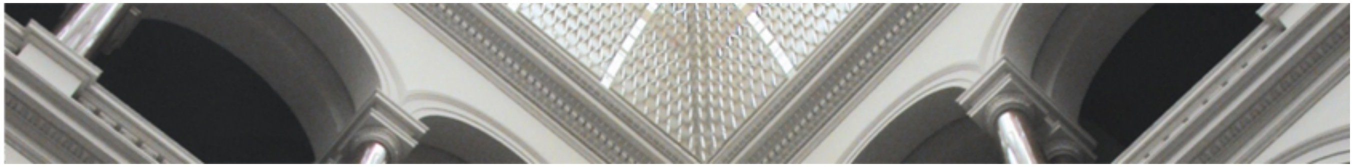
3d Hand Model

- Hand scale change dynamically depending on the input
- All objects/geometry are depending on the palm



```
// Palm /////////////////////////

var costumeshape = new THREE.Shape();

costumeshape.moveTo(1, 1);
costumeshape.bezierCurveTo(1, 5, 20, 10, 20, 1);
costumeshape.bezierCurveTo(20, -35, -1, -15, 1, -10);

var extrudeSettings = { depth: 1.5, bevelEnabled: true, bevelSegments: 10, steps: 1, bevelSize: 1, bevelThickness: 1 };

var geometry = new THREE.ExtrudeGeometry(costumeshape, extrudeSettings);
var Palm = new THREE.Mesh(geometry, material);
Palm.scale.set(palmllength(palmloc,Middleknuckl)*0.1, palmllength(palmloc,Middleknuckl), palmllength(palmloc,Middleknuckl)*0.1);
Palm.position.x = -1;

Palm.add(Indexknuckles);
Palm.add(Middleknuckles);
Palm.add(Ringknuckles);
Palm.add(Littelknuckles);
Palm.add(Thumbknuckles);
```





Achievement in progress

- Preprocessing is done
- Second Phase Training the Model
- Third Phase Evaluation the Model
- Integration of the models in one Desktop Application not a web Application