

Dự đoán cân nặng và chiều cao

Height and Weight Estimation

Nhóm 4

Báo cáo tiến độ

01

Tiến độ

**Trình bày tiến độ công
việc tới ngày 08/11**

02

Kế hoạch

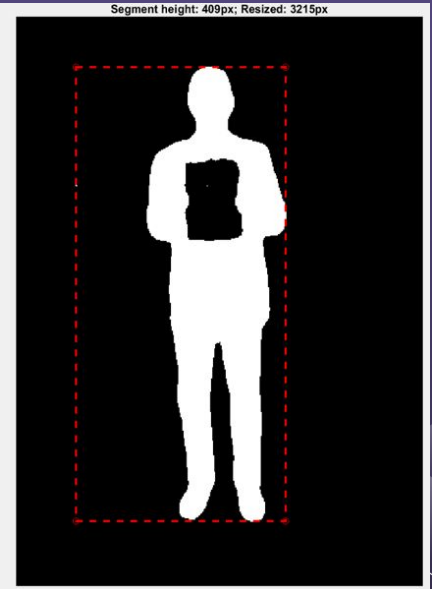
**Kế hoạch trong những
tuần sắp tới**



01

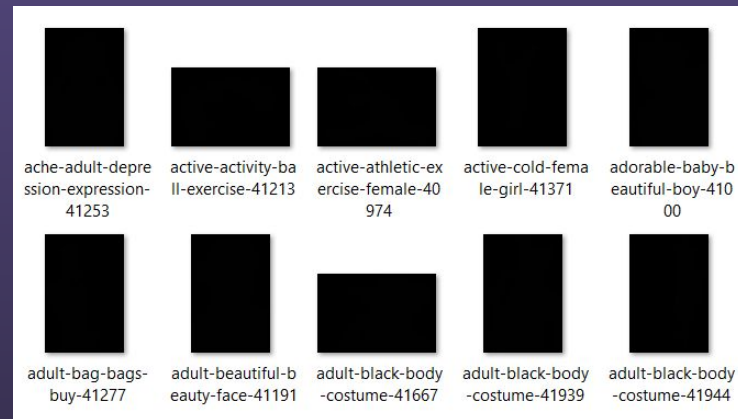
Tiến độ

Height Estimation



1. Training data

- Tập dữ liệu thu thập được gồm 5110 ảnh cho training và 568 ảnh cho validation
- Các ảnh trong tập dữ liệu sẽ được đánh mask theo kích thước cơ thể của người trong ảnh



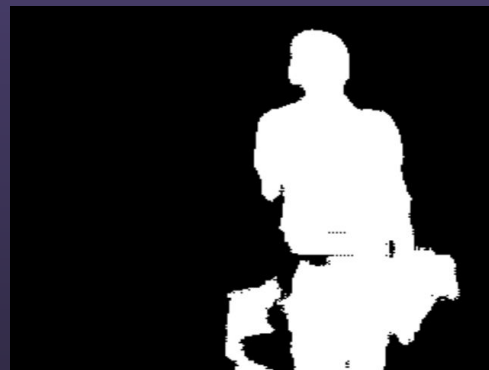
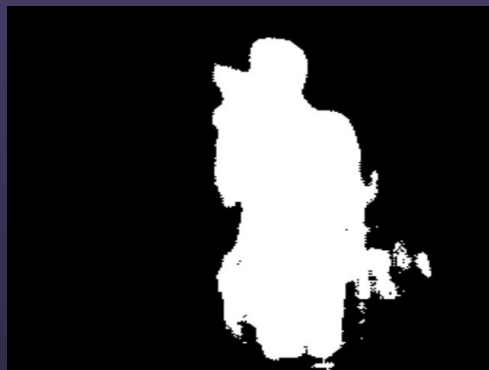
1. Training data

- Sau đó tiến hành training data bằng mô hình mạng UNET
- Sau khi training mô hình thành công ta sẽ thu được file chứa trọng số (weights) đã được huấn luyện của mô hình UNET là unet.h5

```
Epoch 1/5
379/379 [=====] - ETA: 0s - loss: 0.3812 - mean_io_u: 0.3722 - recall: 0.6162 - precision: 0.6899
Epoch 1: saving model to /content/drive/MyDrive/ComputerVision/Project/unet.h5
379/379 [=====] - 1429s 4s/step - loss: 0.3812 - mean_io_u: 0.3722 - recall: 0.6162 - precision: 0.6899 - val_loss: 0.4629 -
Epoch 2/5
379/379 [=====] - ETA: 0s - loss: 0.2956 - mean_io_u: 0.3722 - recall: 0.7130 - precision: 0.7730
Epoch 2: saving model to /content/drive/MyDrive/ComputerVision/Project/unet.h5
379/379 [=====] - 291s 768ms/step - loss: 0.2956 - mean_io_u: 0.3722 - recall: 0.7130 - precision: 0.7730 - val_loss: 0.3431
Epoch 3/5
379/379 [=====] - ETA: 0s - loss: 0.2600 - mean_io_u: 0.3722 - recall: 0.7622 - precision: 0.7987
Epoch 3: saving model to /content/drive/MyDrive/ComputerVision/Project/unet.h5
379/379 [=====] - 302s 796ms/step - loss: 0.2600 - mean_io_u: 0.3722 - recall: 0.7622 - precision: 0.7987 - val_loss: 0.2943
Epoch 4/5
379/379 [=====] - ETA: 0s - loss: 0.2384 - mean_io_u: 0.3722 - recall: 0.7859 - precision: 0.8178
Epoch 4: saving model to /content/drive/MyDrive/ComputerVision/Project/unet.h5
379/379 [=====] - 306s 806ms/step - loss: 0.2384 - mean_io_u: 0.3722 - recall: 0.7859 - precision: 0.8178 - val_loss: 0.2526
Epoch 5/5
379/379 [=====] - ETA: 0s - loss: 0.2195 - mean_io_u: 0.3722 - recall: 0.8104 - precision: 0.8306
Epoch 5: saving model to /content/drive/MyDrive/ComputerVision/Project/unet.h5
379/379 [=====] - 285s 753ms/step - loss: 0.2195 - mean_io_u: 0.3722 - recall: 0.8104 - precision: 0.8306 - val_loss: 0.2246
<keras.callbacks.History at 0x7f969570f730>
```

1. Training data

- Ta sẽ sử dụng file trọng số unet.h5 tiến hành chạy mô hình unet để predict mask như hình dưới

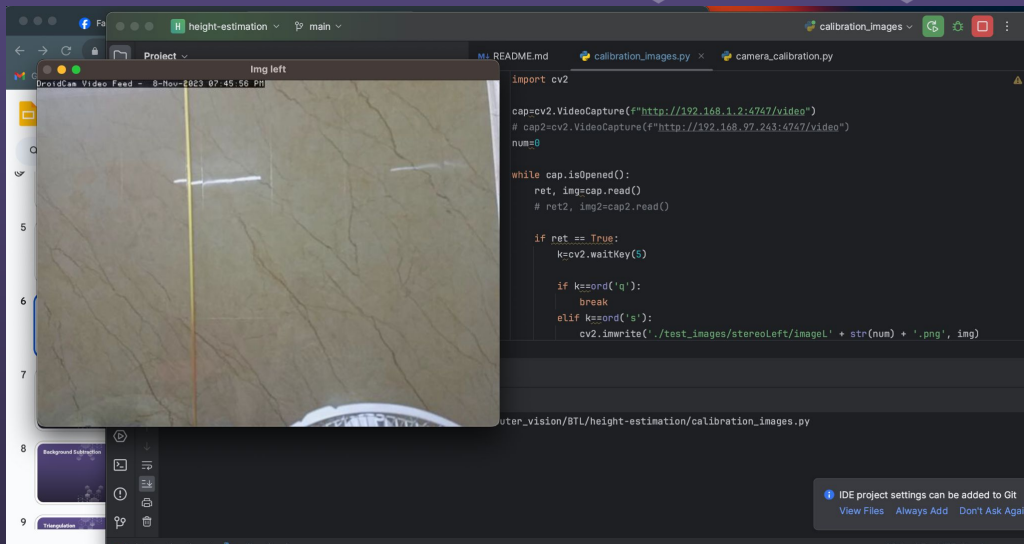


Segmentation với UNet

```
people_segmentation
├── images
│   └── *.jpg # images for segmentation
├── masks
│   └── *.png # segmentation masks corresponds to the images
└── segmentation
    ├── train.txt # image list for training
    ├── trainval.txt # all images
    └── val.txt # image list for validation
```



2. Calibration



Dùng ứng dụng DroidCam trên điện thoại và kết nối với máy tính thông qua IP để thực hiện chụp và lưu ảnh

Calibration Camera

```
# Find the chess board corners
ret, corners = cv2.findChessboardCorners(gray, (width, height), None)

# If found, add object points, image points (after refining them)
if ret:
    objpoints.append(objp)

    corners2 = cv2.cornerSubPix(gray, corners, (11, 11), (-1, -1), criteria)
    imgpoints.append(corners2)

# Draw and display the corners
img = cv2.drawChessboardCorners(img, (width, height), corners2, ret)
cv2.drawChessboardCorners(img, (7, 6), corners2, ret)
cv2.imshow('img', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, gray.shape[:-1], None, None)
```

Calibration Camera bằng
chess board làm corners để
tiến hành hiệu chỉnh

Calibration Images

```
print("Saving parameters!")
param_file = cv2.FileStorage('stereoMap.xml', cv2.FILE_STORAGE_WRITE)
param_file.write('stereoMapL_x', stereoMapL[0])
param_file.write('stereoMapL_y', stereoMapL[1])
param_file.write('stereoMapR_x', stereoMapR[0])
param_file.write('stereoMapR_y', stereoMapR[1])
param_file.write('projMtxL', projMtxL)
param_file.write('projMtxR', projMtxR)
param_file.write('calibrateMtxL', newCameraMtxL)
param_file.write('calibrateMtxR', newCameraMtxR)
param_file.write('distL', distL)
param_file.write('distR', distR)
param_file.write('rot', rot)
param_file.write('trans', trans)
print(retStereo)
```

- Thực hiện Calibration với ảnh bàn cờ
- Tính toán ra các parameters và lưu lại

3. Rectification Images



Để biến đổi hình ảnh thành hình ảnh mới thông qua các parameters của stereoMap đã lưu ở trên



4. Background Subtraction

Thực hiện Background Subtraction với mô hình UNET đã huấn luyện



5. Triangulation

Sử dụng Sift Feature phát hiện các điểm đặc biệt trên hình ảnh, bất kể tỉ lệ và góc nhìn của chúng





02

Kế hoạch



Kế hoạch sắp tới nhóm em sẽ:

- Hoàn thiện, cải tiến Height Estimation
- Thực hiện Weight Estimation



Weight Estimation



Mô hình dự kiến sử dụng kiến trúc mạng nơ-ron sâu, chẳng hạn như mạng nơ-ron tích chập (CNN), để trích xuất các đặc trưng quan trọng từ hình ảnh.





Thanks for watching!

