

# Object Recognition Practical Sessions

Meysam Madadi

[meysam.madadi@gmail.com](mailto:meysam.madadi@gmail.com)

Date	Theory title	Teacher	Practical session
20-02 - 2024	Presentation and CNN basics	Sergio	P1 intro
27-02 - 2024	CNN and GNN architectures	Meysam	-
05-03 - 2024	Object detection and segmentation	Meysam	P1 Q&A - P2 intro
12-03 - 2024	Human pose estimation	Meysam	P1 presentation
19-03 - 2024	Human behaviour understanding	Sergio	-
26-03 - 2024	Easter holidays	-	
02-04 - 2024	INVITED TALK		P2 Q&A - P3 intro
09-04 - 2024	Exams week	-	
16-04 - 2024	Master seminar	-	
23-04 - 2024	Recurrent models and transformers	Meysam	P2 presentation
30-04 - 2024	Presentation I	Sergio	
07-05 - 2024	Presentation II	Sergio	
14-05 - 2024	Generative models	Meysam	P3 Q&A
21-05 - 2024	MONDAY schedule	-	
28-05 - 2024	Exam	Sergio	P3 presentation

7 sessions  
3 blocks

# Deliverables

1. CNN architecture design, **deadline 11/03/2023 23:59**
2. Fashion parsing (segmentation), **deadline 22/04/2023 23:59**
3. Body and clothes depth estimation, **deadline 27/05/2023 23:59**

## Notes:

- The tasks must be done in the groups of **4** students. One student in each group must send me the name of his/her mates in the group in one week. If I do not receive this email, I will assign students to the groups randomly.
- Deliverable is a zip file including the code, report pdf and presentation slides.
- The report must fit within the **10** page limit using the font **calibri** with size **11**. Extra pages will be penalized.
- Submissions must be done through virtual campus.
- Submissions after the deadline will be penalized.
- The solutions and results must be presented by the members of each group in 10 minutes on the defined days,
- Deliverables are evaluated based on the quality of the report and presentation, and the number of tasks done. Erroneous results without proper justification won't be taken into account.

# Deliverable 1 - CNN architecture design

- The goal is to learn how to design and train CNN networks to maximize the performance,
- The task is the multi-label classification on the [Pascal VOC 2012 dataset](#),
- You can use the given code in the practical session as your base code.

# Deliverable 1 - Tasks

1- Analyze the dataset for the distribution of the labels by:

- a. Counting the objects;
- b. Counting the images, that is no matter how many objects of the same label appear on the image, all counts 1;
- c. Mean and variance of the area ratio (dividing object area to image area) of objects per label. You can use object bounding box to compute the area.

# Deliverable 1 - Tasks

2- Train and evaluate the **ResNet50V2**, **EfficientNetV2B0** and **ConvNeXtTiny** from scratch vs pretrained on imagenet using the default hyperparameters and classifier head given in the practical notebooks. Note that the training and evaluation must be done on the given train/test txt files. Then select the best performing network and do the following tasks incrementally:

- a. Tune the batch size using this list (as far as memory allows) [16, 32, 64, 128].
- b. Develop two data augmentation algorithms of your choice along with the default ones in the given notebook.
- c. The dataset is unbalanced. Develop two algorithms of your choice to have a better balance among labels during the training without sacrificing the accuracy.
- d. Redesign the classifier head to have better results.
- e. Increase the number of epochs as much as needed to maximize the test set performance.

Write the report by explaining, comparing and analysing the results.

# Deliverable 2 - fashion parsing

What do you need to do in this task?

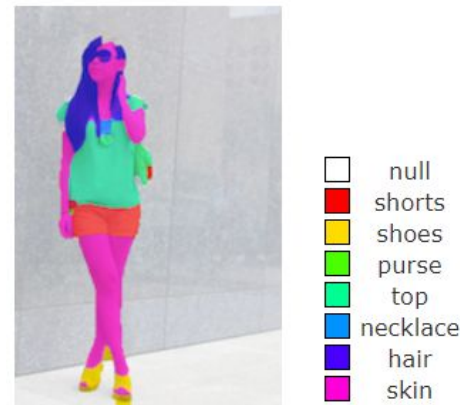
- Fashion semantic segmentation

On which dataset?

- Fashionpedia: <https://fashionpedia.github.io/home/>

Using which code base?

- MMSegmentation: <https://github.com/open-mmlab/mmdetection>
- Or any code you find more convenient

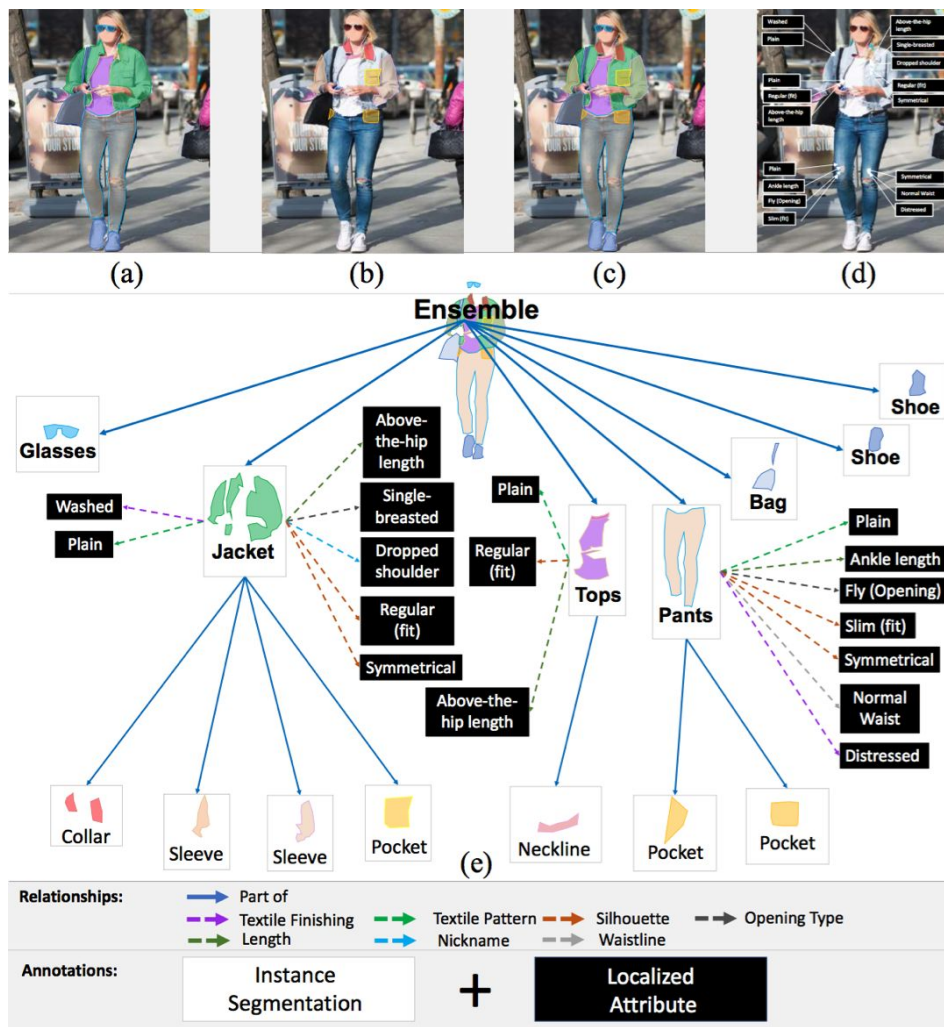


Clothing parsing

# Fashion parsing

Hierarchical segmentation  
and attribute detection

- We do not consider item parts and attributes,
- Only the person in the middle has been annotated.
- We use the validation set to report the results.





# Deliverable 2 - Tasks

1- Analyze the dataset for the distribution of the labels by:

- Counting the number of images per label,
- Mean and variance for the pixel ratio per label:
  - $(\# \text{ of pixels of label } j \text{ in image } i) / (\# \text{ of pixels in image } i),$
  - $(\# \text{ of pixels of label } j \text{ in image } i) / (\# \text{ of } \mathbf{foreground} \text{ pixels in image } i) \implies \text{in this case we exclude the background}$

# Deliverable 2 - Tasks

2- Choose three segmentation networks (justification is required) from the MMSegmentation implementations and finetune them on the fashionpedia dataset, as far as your resources allows, as following:

- Use the default learning rate as suggested by MMSegmentation,
- Tune batch size based on your resources,
- Apply the basic data augmentation like rotation, scaling, cropping, etc,
- Train based on 192 vs 384 px image resolution. Is there any relationship between the resolution and accuracy of each label?

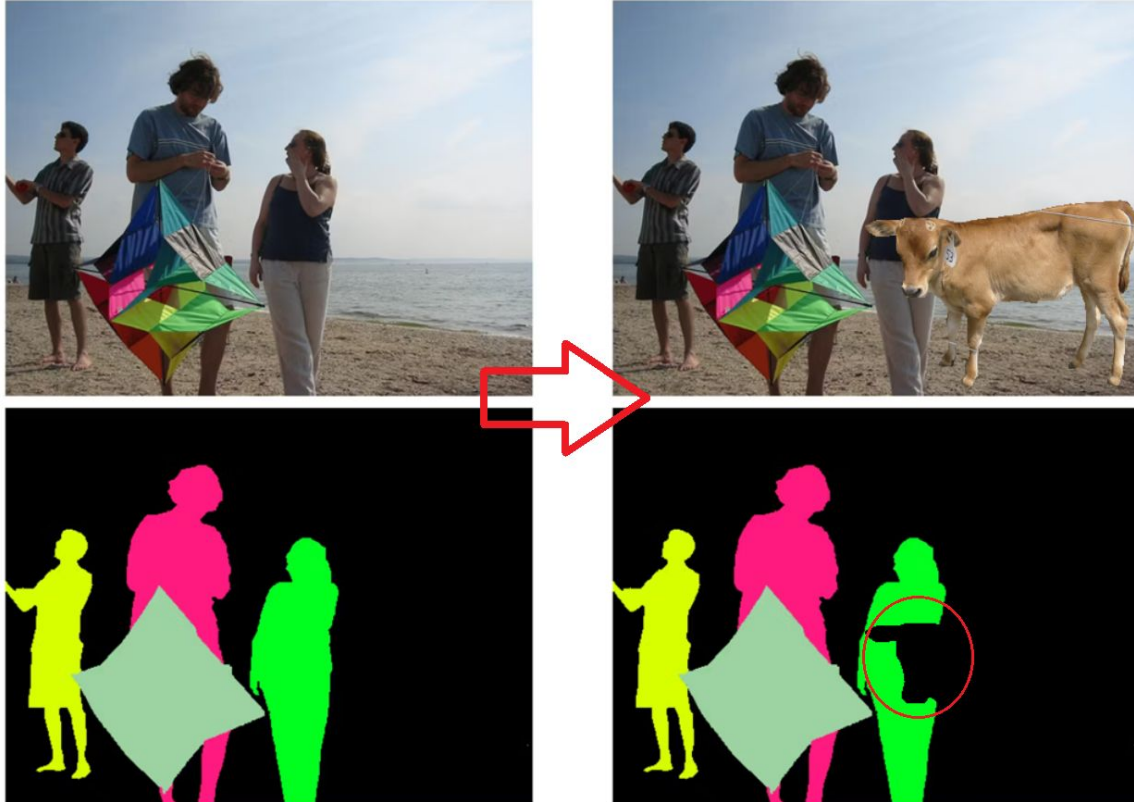
Select the best performing network and resolution and do the following:

- Discard the large and overrepresented labels and train the network with the same hyperparameters as above. Does this make any difference to the results of small objects?

# Deliverable 2 - Tasks

3- Implement the following data augmentation and train the best performing network:

- Using the instance segmentation annotations of PASCAL VOC dataset, copy object instances randomly and paste them in random locations on the target people in fashionpedia images while updating the ground truth segments.



## Deliverable 2 - Tasks

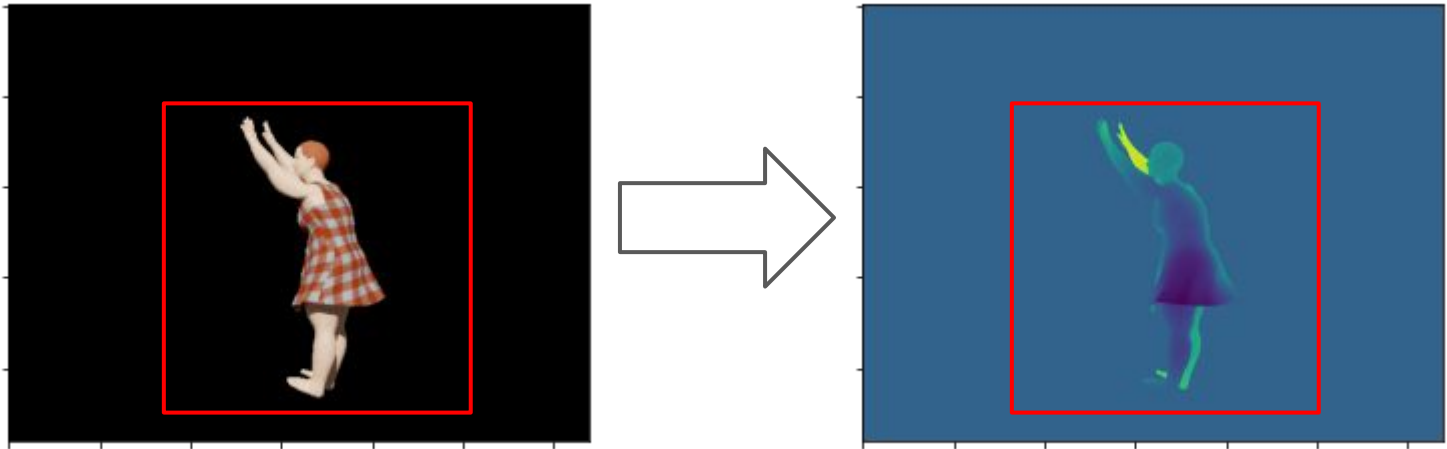
4- Without implementing any code or training, just discuss two possible ideas to improve the results like network modification, higher order relationships as extra heads, losses, leveraging unlabeled data, etc

NOTES for writing the report:

- Justify your choices,
- Thoroughly discuss the results both qualitatively and quantitatively,
- Report the results using accuracy and mDice metrics excluding the background,
- Your best performing network must show at least 20% mDice

## Deliverable 3 - Body and cloth depth estimation

- Given an image of a person, cropped in the preprocessing (red rectangle), the goal is to estimate its corresponding depth image.



# Deliverable 3 - Body and cloth depth estimation

- Dataset:

- A subset of the CLOTH3D++ dataset (<https://chalearnlap.cvc.uab.cat/dataset/38/description/>), could be downloaded from [cloth3d++\\_subset.zip](#). The first 128 folders (00001 to 00152) must be used for training, the following 16 folders (00153 to 00168) for validation and the rest for test.
- You can play with the data, see data structure, visualize the 3D, render depth images and extract frames from the RGB videos using the given starter kit.
- If you use colab, the content of the starter kit must be uploaded to your google drive under “cloth3d” folder.

- Preprocessing:

- Crop and save the images such that 1) the center of the subject and cropping to be the same, 2) leave 10px margin between cropping and subject boundaries, and 3) apply square cropping. NOTE: in some frames the subject may go out of the scene. You can ignore these frames, for instance by counting the number of pixels in the mask and thresholding. The frames must be saved as .jpg under “image” folder with naming protocol “<folder name where the video is located>\_<frame number>”
- Rendered depth must be cropped similar to RGB frame and saved as .npy (using numpy.save) under “depth” folder with the same naming protocol as RGB frames.
- The cropped images can be resized to 256x256 before saving them. Note that resizing the depth images could be tricky especially for the boundaries. You can use nearest neighbor values to avoid the effect of the zero background.

# Deliverable 3 - Body and cloth depth estimation

- A simple baseline code is given for depth estimation using UNET architecture.
- Your job is:
  - preprocessing the data,
  - Training the baseline and improving the results by the following ways:
    - Applying data augmentation relevant to the problem,
    - Tuning some of the hyperparameters of your choice. No need for a massive grid search. Just try a few that makes more sense to you.
    - Training with better losses. You can inspire from here: [https://keras.io/examples/vision/depth\\_estimation/](https://keras.io/examples/vision/depth_estimation/). NOTE: consider the multiplication of the prediction with the gt mask before computing the loss. We can assume this mask is always available.
    - Studying the impact of surface normals in depth estimation, e.g. predicting normals along with depth or predicting normals first and using them to predict depth,
    - Studying the vision transformer architectures by replacing the unet backbone. Try no more than 3 options.
  - You can perform the above studies incrementally. For example, you can study depth normalization on top of baseline network and data augmentation.
- Thoroughly discuss the results both quantitatively and qualitatively.

# Body and cloth depth estimation

Tips and modifications required on the code

For a faster I/O operation, you may

1- Train with multiple workers, e.g. `model.fit(....., workers=4)`, or

2- (optional) Save the whole data in a tfrecord file and iterate over it, an example here:

[https://keras.io/examples/keras\\_recipes/tfrecord/](https://keras.io/examples/keras_recipes/tfrecord/)