Object Recognition
Practical Sessions

Meysam Madadi

Date	Theory title	teacher	Practical session
02/22	Presentation and CNN basics	Sergio	CNN basics I
03/01	Backbone architectures	Meysam	CNN basics II
03/08/2022	Recurrent architectures	Sergio	CNN advanced architectures
03/15/2022	Object detection and segmentation	Meysam	Object detection I
03/22/2022	Human pose estimation	Meysam	Object detection II
03/29/2022	Human behaviour	Sergio	Human pose I
04/05/2022	Exams week (31/03 - 06/04)	-	-
04/12/2022	Easter holidays (11/04 - 18/04)	-	-
04/19/2022	Presentation I	Sergio	-
04/26/2022	Transformers	Meysam	Human pose II
05/03/2022	Graph neural networks	Meysam	Object recognition
05/10/2022	Master seminar (09/05 - 13/05)	-	-
05/17/2022	Presentation II	Sergio	-
05/24/2022	Exam		

8 sessions
3 blocks

Deliverables

1. Contextual data augmentation, deadline 21/03/2022 23:59
2. Fashion parsing (segmentation), deadline 25/04/2022 23:59
3. Body and clothes depth estimation, deadline 03/06/2022 23:59

Contextual data augmentation

1. Select one of the networks studied in the class,
2. Train the network on Pascal VOC dataset for multi-label classification,
3. During the training corrupt the training images with contextual data augmentation,
    a. Select random objects and put them on random locations in the training image.
4. Study the results. What happens if
    a. No contextual data augmentation is applied,
    b. objects overlap vs no overlapping,
    c. objects appear in random scales and orientation,
    d. objects are selected such that the whole dataset is balanced, i.e. the number of labels in the whole dataset is equal,

Contextual data augmentation

- The report and code/s must be uploaded to the virtual campus before the deadline.
- The report must be short. Maximum 2 pages with font size 11.
- The report must at least contain the following information:
  - Which network has been used and why,
  - How the network has been trained: hyperparameters, optimizer, loss, training strategy, etc,
  - A thorough discussion of the results.

Fashion parsing

It can be defined as one of these problems:

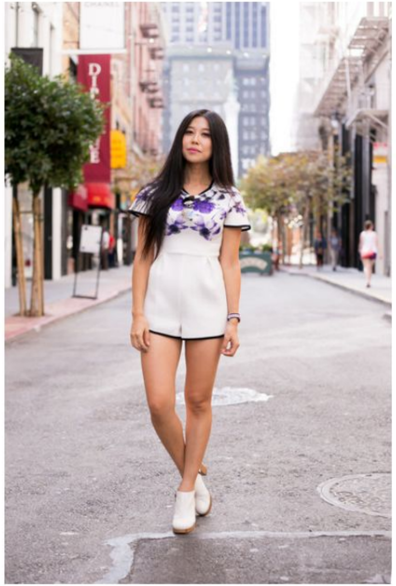
- Fashion attributes classification,
- Fashion description by caption,
- Semantic segmentation,
- Hierarchical segmentation and attribute detection

Fashion parsing

Fashion attributes classification

Fashion parsing

Fashion description by caption



LOS ANGELES, CA

466 FANS
288 VOTES
62 FAVOURITES

TAGS

CHIC
EVERDAY
FALL

COLOURS

WHITE-BOOTS

NOVEMBER 10, 2014

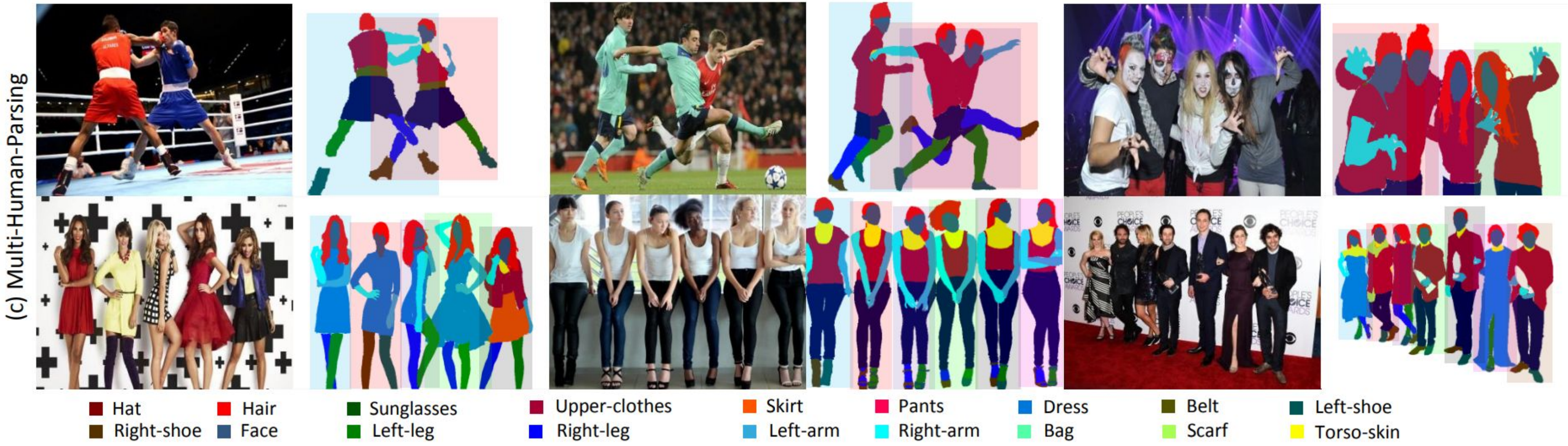
GARMENTS

White Cheap Monday Boots
Chilli Beans Sunglasses
Missguided Romper
Daniel Wellington Watch

COMMENTS

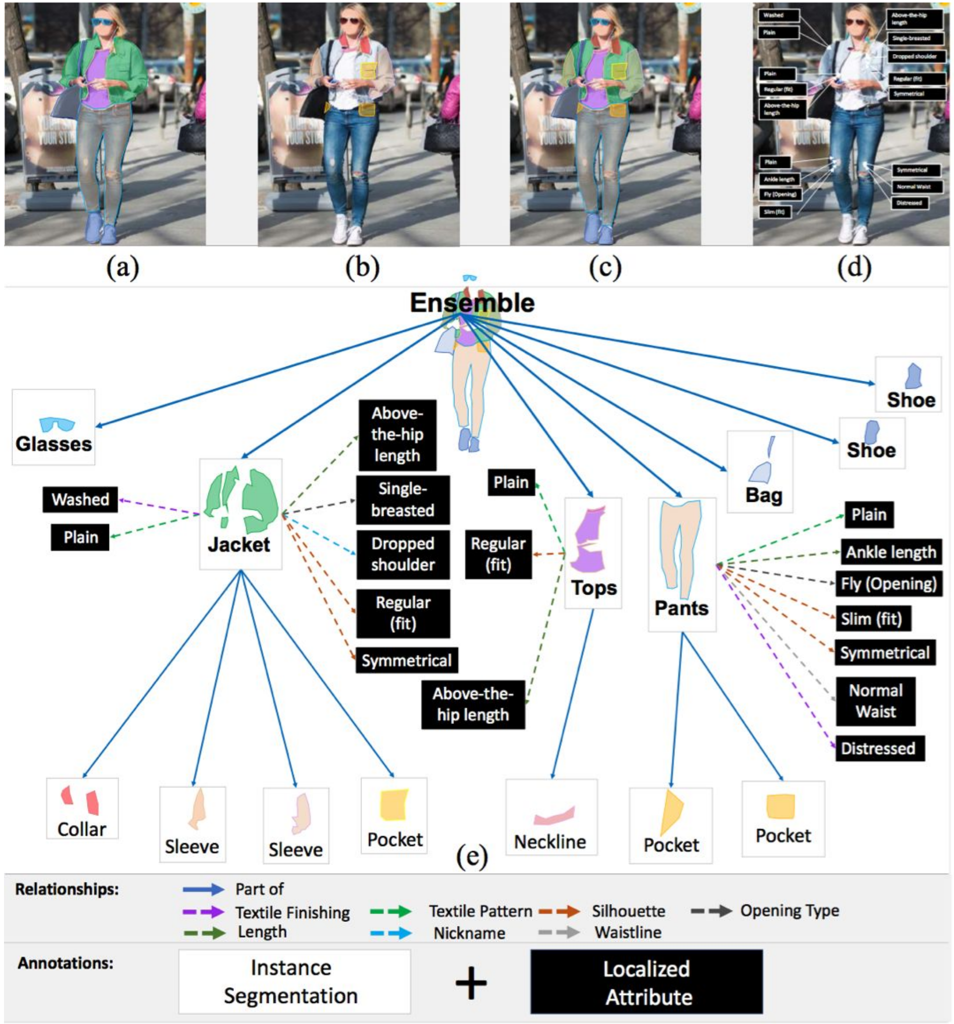
Nice!!
Love the top!
cute
...

Fashion parsing
Semantic segmentation



Fashion parsing

Hierarchical segmentation and attribute detection



Fashion parsing

What do you need to do in this task?

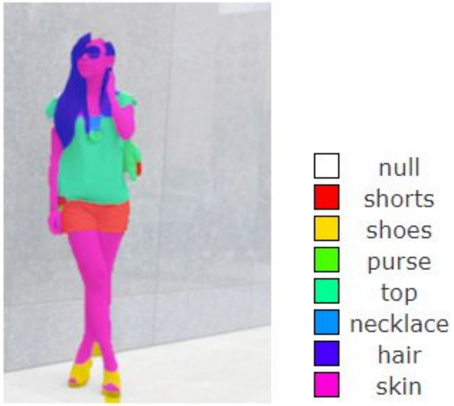
- Fashion semantic segmentation

On which dataset?

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

How?

- Select a segmentation algorithm from https://paperswithcode.com/task/semantic-segmentation

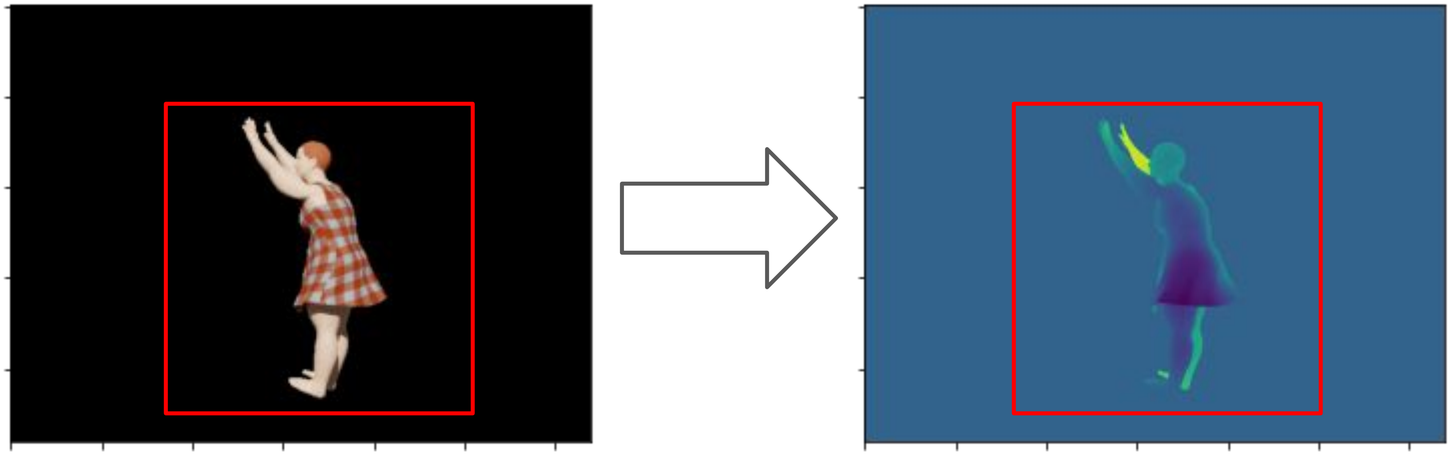


Clothing parsing

Fashion parsing

- The report and code/s must be uploaded to the virtual campus before the deadline.
- Students must confirm the selected algorithm by email before **APR 5th.** It is recommended to assure the code is stable and works properly before this date.
- The report must be short. Maximum 3 pages with font size 11.
- The report must at least contain the following information:
  - A short summary of the selected algorithm and justification why it is selected,
  - How the network has been trained: hyperparameters, optimizer, loss, training strategy, etc,
  - A short summary and statistics of the dataset,
  - A thorough discussion of the results.

Body and cloth depth estimation



Body and cloth depth estimation

- Dataset:
  - A subset of the CLOTH3D++ dataset (https://chalearnlap.cvc.uab.cat/dataset/38/description/). Randomly select **4K** images as the test data. The link to download the data: cloth3d++_subset.zip
- Depth rendering:
  - Based on the code given for PR7.
- 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 ot the scene. You can ignore these frames.
- Training model:
  - UNET based on this code: https://keras.io/examples/vision/depth_estimation/

Body and cloth depth estimation

- The report and code/s must be uploaded to the virtual campus before the deadline.
- The report must be short. Maximum 3 pages with font size 11.
- The report must at least contain the following information:
  - How the network has been trained: hyperparameters, optimizer, loss, training strategy, etc,
  - A short summary of the dataset,
  - A thorough discussion of the results including:
    - The impact of image resolution on the results (256 vs 128),
    - The impact of contextual data augmentation, using PASCAL VOC 2007 objects with at most two random objects per image, random rotation and scale, and overlapping allowed.
    - Tuning the loss functions.

Body and cloth depth estimation

Tips and modifications required on the code

In the python notebook of PR7:

```
thresh = max_depth - 1
mask = depth<thresh
dmin = depth.min()
dmean = depth[mask].mean()
dmax = depth[mask].max()
depth[mask] = depth[mask] - dmean
depth[~mask] = 0.0
```

```
thresh = max_depth - 1
mask = depth<thresh
dmin = depth.min()
dmean = depth[mask].mean()
dmax = depth[mask].max()
depth[mask] = (depth[mask] - dmin) * 1000 + 1
depth[~mask] = 0.0
```

Body and cloth depth estimation

Tips and modifications required on the code

```
def load(self, image_path, depth_map, mask):
    """Load input and target image."""

    image_ = cv2.imread(image_path)
    image_ = cv2.cvtColor(image_, cv2.COLOR_BGR2RGB)
    image_ = cv2.resize(image_, self.dim)
    image_ = tf.image.convert_image_dtype(image_, tf.float32)

    depth_map = np.load(depth_map).squeeze()

    mask = np.load(mask)
    mask = mask > 0

    max_depth = min(300, np.percentile(depth_map, 99))
    depth_map = np.clip(depth_map, self.min_depth, max_depth)
    depth_map = np.log(depth_map, where=mask)

    depth_map = np.ma.masked_where(~mask, depth_map)

    depth_map = np.clip(depth_map, 0.1, np.log(max_depth))
    depth_map = cv2.resize(depth_map, self.dim)
    depth_map = np.expand_dims(depth_map, axis=2)
    depth_map = tf.image.convert_image_dtype(depth_map, tf.float32)

    return image_, depth_map
```

Remember to save the depth images as numpy arrays.

Remove the red box

Body and cloth depth estimation

Tips and modifications required on the code

```
def call(self, x):
    c1, p1 = self.downscale_blocks[0](x)
    c2, p2 = self.downscale_blocks[1](p1)
    c3, p3 = self.downscale_blocks[2](p2)
    c4, p4 = self.downscale_blocks[3](p3)

    bn = self.bottle_neck_block(p4)

    u1 = self.upscale_blocks[0](bn, c4)
    u2 = self.upscale_blocks[1](u1, c3)
    u3 = self.upscale_blocks[2](u2, c2)
    u4 = self.upscale_blocks[3](u3, c1)

    return self.conv_layer(u4)
```

Multiply the output with the groundtruth subject mask

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/