1 The Dataset

To perform the pose estimation, we need some data to train, validate and test our model. Throughout this section the data used is described and preprocessed.

1.1 The COCO Dataset

Figure 1: Example of image from the COCO dataset with labels

Notice how the image contains multiple people, each with their own keypoints and amount of joints labeled

The data needed for our model has to fit our problem and has to be annotated, as our model will perform supervised learning. There are multiple datasets that fits these requirements - one of these datasets is Common Objects in Context (COCO) dataset [3], which we will use. The dataset contains annotations for different purposes, however, we only need the keypoint annotations of human bodies. An example of such a picture with the labels can be seen in Figure 1.
The annotation of each person consists of an array with a length of 51. Each joint corresponds to three sequential elements in the array, where the first index tells the x-location of the joint in the image, the second index tells the y location of the joint in the image, and the third index is a flag, v, telling the visibility of that joint in the image. v has three outcomes; if $v = 0$, then the joint is not labeled, if $v = 1$, then the joint is labeled but not visible, and if $v = 2$, then the joint is visible and labeled.
The dataset is split into three parts; a part used for training the model, a part used for validating the model and a part used for testing the model. However, the part used for testing the model is unlabel, hence, why it is unusable for our purpose, as our model will be doing supervised learning, where the labels are needed. As both the training dataset and the validation dataset will be used for training and tuning the model, we will need to create our own dataset for testing to provide an unbiased evaluation of the final model [1].
The training and validation sets contains a total of about 123.000 various images. As we only need the images that contain humans, we will be discarding the images without any humans, leaving us with a total of about 66.808 images of humans doing various tasks. Each image can contain multiple people, which we need to handle before training our model, as we will be focusing on single-human pose estimation. Besides this, each image also has different resolution and aspect ration, which we also need to handle, as our model requires the images to have a fixed resolution.
Lastly, we should also do some handling of the labels before training the model for two reasons

1. There could have been some inaccuracies, when the joints were labeled (especially when $v = 1$; when the joint is labeled but not visible)

2. Each joint could correspond to multiple pixels in the image, hence why it is not correct to only use a single pixel as the location of the joint in the image (as it is the case currently).

1.2 Data Preprocessing

1.2.1 Creating the test dataset

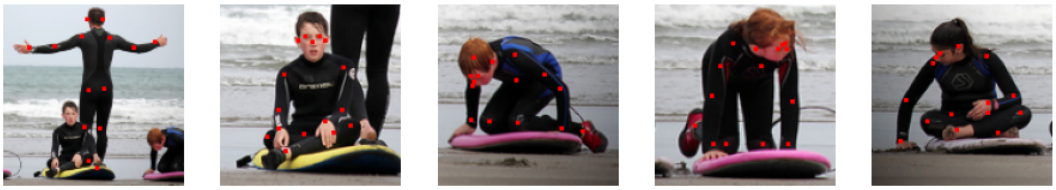
To create the dataset which will be used for testing, we take the training set, since it is the larger of the training set and the validation set, and sample randomly 20% of the images from it without replacement, to create a test set. This new test set will not be used when training the model, nor used when tuning the parameters. Instead, it will only be used when evaluating the final model.

1.2.2 Preprocessing each image

Figure 2: Data distribution

	Amount of images	Percentage
Training set	x	x
Validation set	5.377	x
Testing set	25.409	x
Total	**x**	**x**

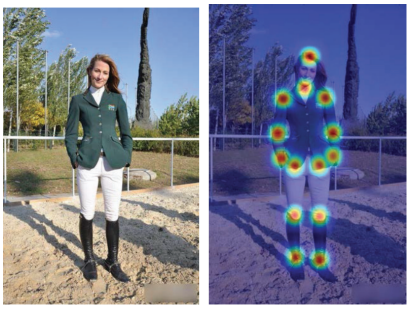
Figure 3: The results of processing the image from Figure 1 with the corresponding labels



When preprocessing the images, we need images containing only a single person and each of the final images needs to have the same dimension. This is done by creating multiple bounding boxes, where each bounding box surrounds a single person, which is done by looking at the outer keypoints for each person. Then, each bounding box is transformed into a square by making the shorter sides have the same length as the longer sides - this is done to keep the aspect ratio of the image, when it is later resized. Since each keypoint does not necessarily lie on the edge of the person, the current bounding boxes would result in not all of the corresponding person being in each the bounding box. For this reason, each bounding box is expanded with 50 pixels in the height and width (25 pixels in each direction). If, however, the image is not big enough to contain the expanded bounding box, then the bounding box is expanded just enough to be contained in the image, while still being a square. Another outcome after the expanding of the bounding box is, that one of the corners of the bounding box lies outside of the image. This is handled by moving the bounding box inside the image by moving it as little as possible.
When all of the above is done, the image is then croped to each bounding box, resulting in multiple squared images, each containing an unique person. Each of these squared images are then resized to a 128×128 image, which is then saved as a `png`-file. Doing all of these steps results in the distribution of images displayed in Figure 2. In Figure 3 the results of processing the image from Figure 1 is shown with the corresponding labels.

1.2.3 Handling the labels

Figure 4: An example of the heatmaps of a single image fused together and put over the original image [2]



Left: The original image. Right: The heatmaps of all the keypoints, fused together to a single image.

For each image of a single person, we create 17 heatmaps, one for each possible joint in the image, which tells the probability of the joint being in each pixel. Such heatmaps can be seen in Figure 4.
The heatmap of a single joint is created firstly, by initializing a all-zero 2D array with size 128×128. Next, in the 2D array at position x, y, corresponding to the position of the joint, a 1 is placed - this 1 now corresponds to where the joint is placed in the image according to the keypoint annotation. Lastly, a Gaussian filter is used to smear out the image, where the standard deviation depends on the visibility of the joint; if the joint is visible, then the standard deviation is 0.5, whereas the standard deviation is 1 if the joint is not visible, since we are more unsure if the joint has been labeled correctly. We do this for all of the 17 joints for each image, resulting in the keypoints which will be used for developing our model.