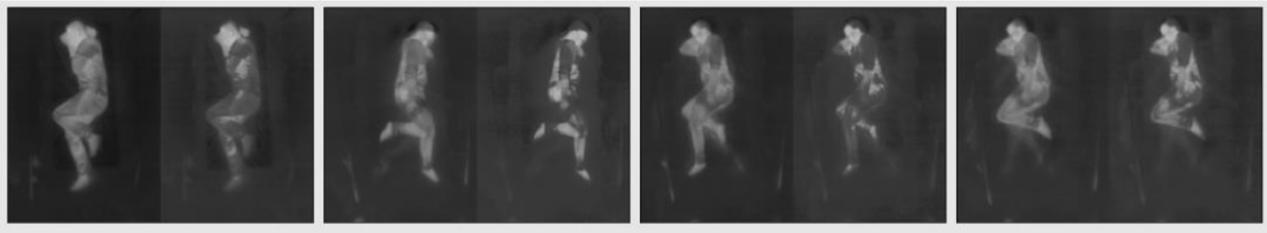
Team Proletarians

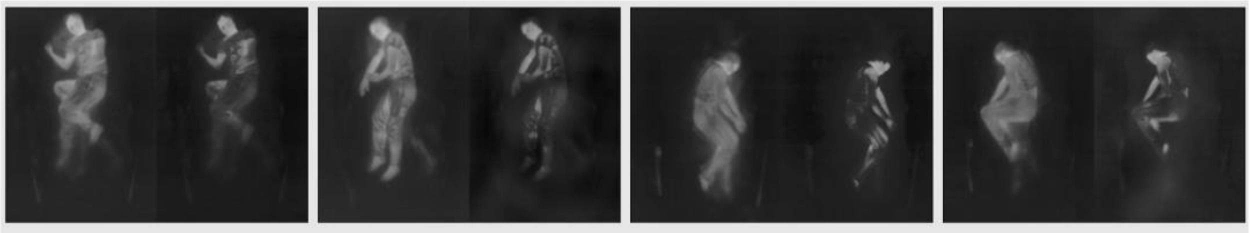
Augmentation

1. **Synthetic Cover Boundary**:

Creating a synthetic cover requires changing the pixel values of a certain area of the IR image so that it matches the original cover images. With that goal in mind, we applied a histogram matching method which adjusted the histogram of the uncovered images to be similar to the histogram of covered images.

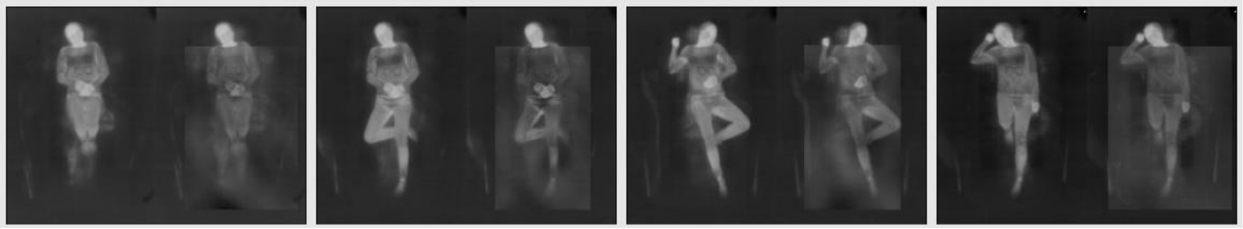


Furthermore, we also applied a Fourier domain adaptation method (FDA) which tried to imitate the frequency distribution of the covered image in the uncovered images. Added on along with the previous augmentation, our synthetic images resembled the original covered images more.



The augmentations made the uncovered images closely resemble the covered images. But in some cases, our augmentation affected the areas outside the cover region. So we limited our augmentation to only a certain part of the image,excluding the head. This was done utilizing the keypoint information available to us.

By confining our augmentations on the region that represents cover, we imitated the sharp change in pixel values transitioning from uncovered regions to covered regions in an original covered image.

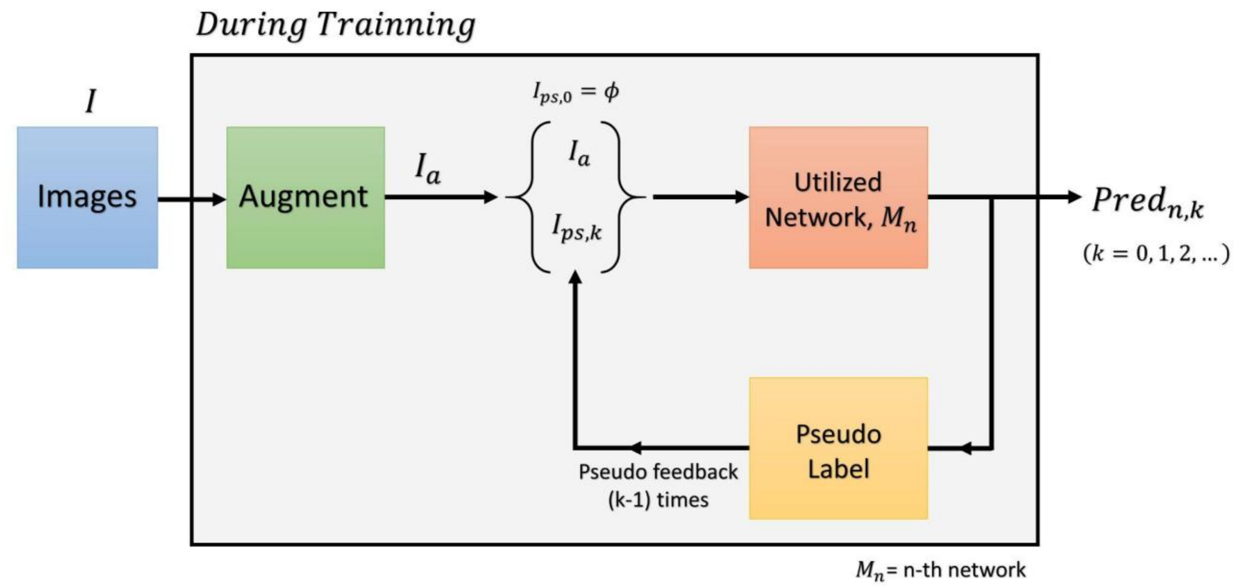


2. **Typical Augmentations**:

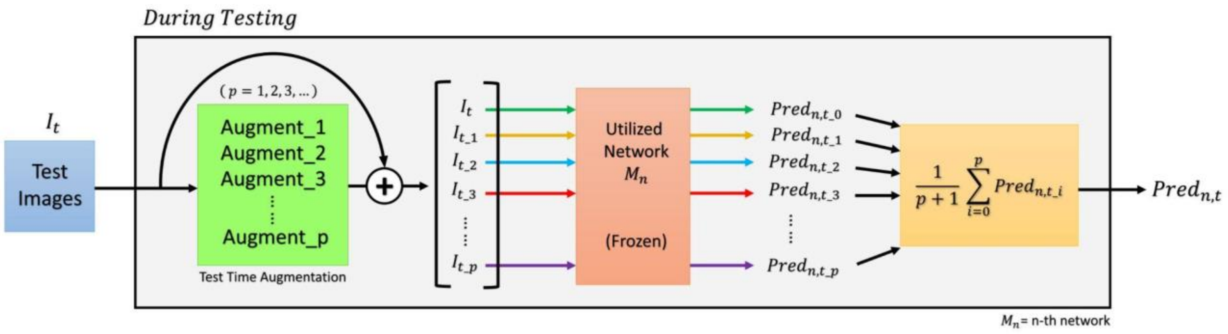
We also made use of the typical augmentations used in deep learning based tasks. Specifically, we used augmentations such as random flip, random rotate, random scaling, random brightness and contrast modification. While using flip based augmentations, the changes in keypoint were taken into account. These augmentations primarily helped in preventing overfitting to the train data, though the changes in brightness and contrast made the model more robust in handling covered cases too. The intensity of the augmentations were kept within reasonable limits to reflect real world cases.

❖ Implemented Prediction Scheme

During training, images get augmented and pass through a network M_n used for prediction. The unlabelled data are predicted using the labelled ones and the obtained predictions are fed back to the network. Therefore, the network now has the original labels as well as the 'pseudo labels' to be trained upon. The process is done 'k'-times among which new pseudo labels are created (k-1) times.



During testing, however, it is only required that the direct prediction is obtained and the weights of the network are kept frozen. But to maximize the score, test time augmentation is implemented. In this way, the mentioned augmentations are done on each image, but not all together; at random. After that, each image, along with its augmented counter-parts, is passed through the same frozen model and the predictions are averaged.



Below we describe the implemented networks within our scheme as well as briefly summarize the pseudo labeling position of our method:

1. **Utilized Network**: For modeling such a network, we have taken help of three different approaches. The used techniques are:

[For using LiteHrnet and TransPose we generated a Bounding Box Using YOLO V5. As for such Top-Down Models, these Bounding Boxes are used to first localize the entire human body using the IR and then it goes towards predicting the key-points.]

- **EvoPose2D**: EfficientNetB5 is used as the main architecture of this model. In the EvoPose2D model, we are taking images of size 512x384 as input and the generated heat-maps are of size 256x192. For loss calculation we are using the MSELoss function and Adam as an optimizer. Here the WarmupCosineDecay function is chosen as the scheduler. In our model the images go through different augmentation processes named varying brightness, contrast, hue, saturation and flipping, rotation as well as we apply affine transformation.

  William McNally, Kanav Vats, Alexander Wong, John McPhee, EvoPose2D: Pushing the Boundaries of 2D Human Pose Estimation using Neuroevolution
  [Reference: https://arxiv.org/pdf/2011.08446.pdf]

- **LiteHrnet:** In the LiteHrnet model, we are taking images of 288x384 size and generating heatmaps of size 72x96. For loss calculation purposes we are using the JointMSELoss function. We have chosen Adam as an optimizer. In our model, images are gone through an augmentation process where we apply HalfBody Transformation, RandomScale Rotation, RandomFlip and Affine Transformation. Git repository name ‘mmpose’, model name ‘LiteHrnet’[https://github.com/open-mmlab/mmpose?fbclid=IwAR1ncsBOUcpJxVjBLYYXxiylQ8uoJOJmTg_SFVMTJ4npsg4mj13GTmOW2p8]

- **TransPose**: In our model we are taking images of size 256x192. We are using two variants of TransPose: R and H variants. In R model architecture, 4 encoder layers and 8 heads in attention layers are used and in H model architecture 4 encoder layers and 1 head in attention layers are used. The generated heatmap size in the model is 64x48 and for loss calculation purposes we are using the JointMSELoss function. CosineAnnealingLR is used as a scheduler. For augmentation purposes we choose to vary brightness, contrast, hue, saturation and apply affine transformation.

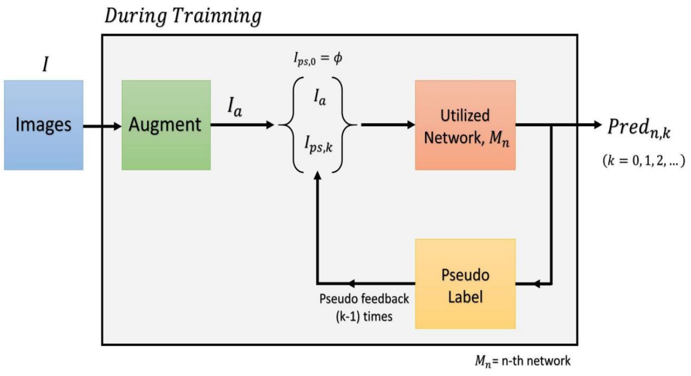
We have omitted the R variant during ensemble, since it deteriorates the overall performance score.

Sen Yang, Zhibin Quan, Mu Nie, Wankou Yang, TransPose: Keypoint Localization via Transformer.
[Reference: https://arxiv.org/pdf/2012.14214v3.pdf]

2. **Pseudo Labeling**: At first, the labeled data is trained in the model. This gives the model a somewhat idea about our dataset at hand. Then the unlabelled data is given as input for labeling. Thus a mixture of unlabeled data and pseudo-labeled data is found. This is known as pseudo-labeling.

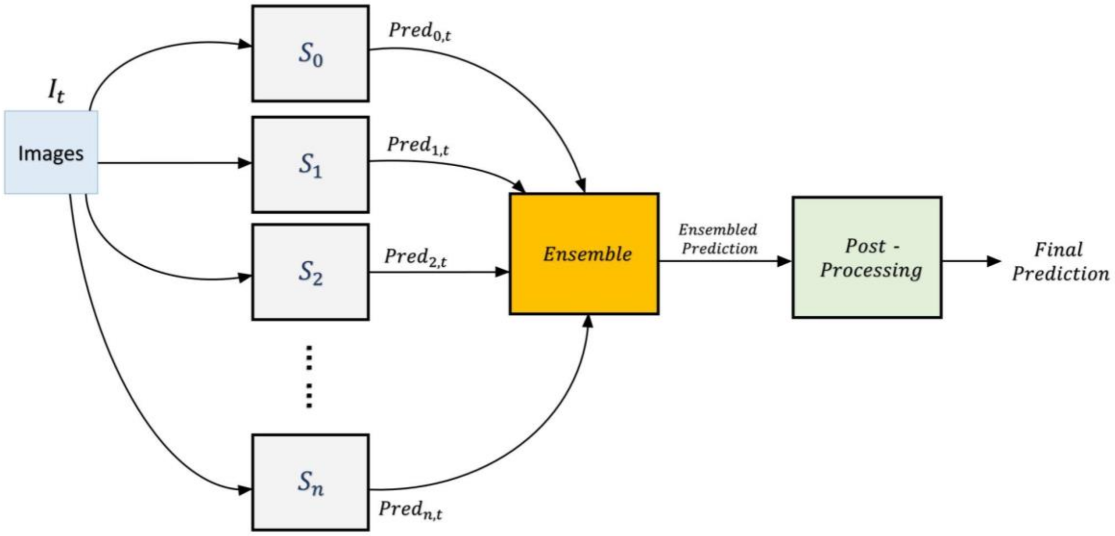
Our scheme's $I_{ps,k}$ is updated in every epoch. The given dataset in the competition contains lots of images. The images can be categorized into three categories such as, uncovered, cover 1 and cover 2. Most of the images belong to cover 1 and cover 2 categories which are unlabeled. Hence, pseudo-labeling is used here.

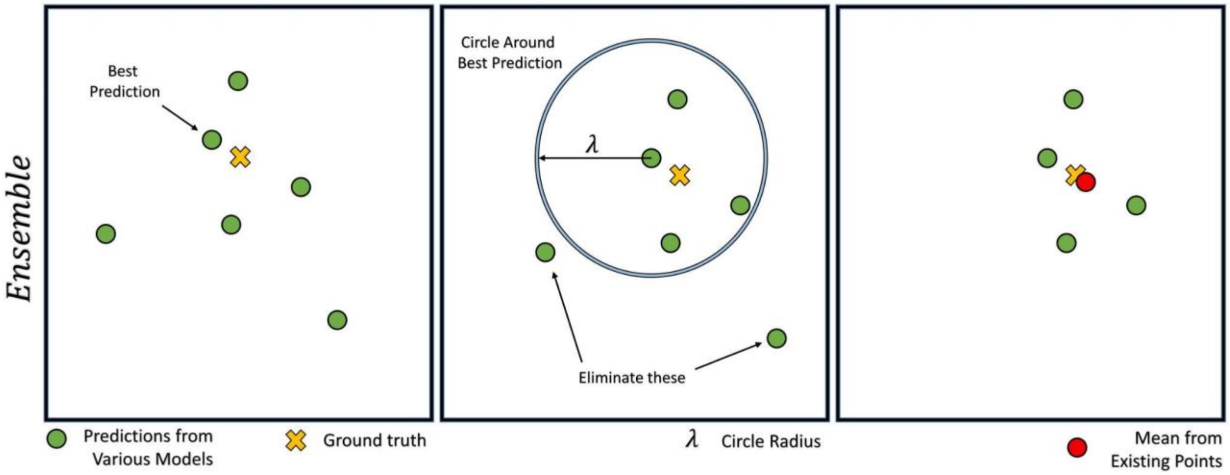
Our model takes the labels of the uncovered images and trains with them. Then these cover 1 and cover 2 images are labeled through machine generation. At first, the labels are full of errors. However, as the training progresses the labels converge into satisfactory labels.



❖ Ensemble Technique

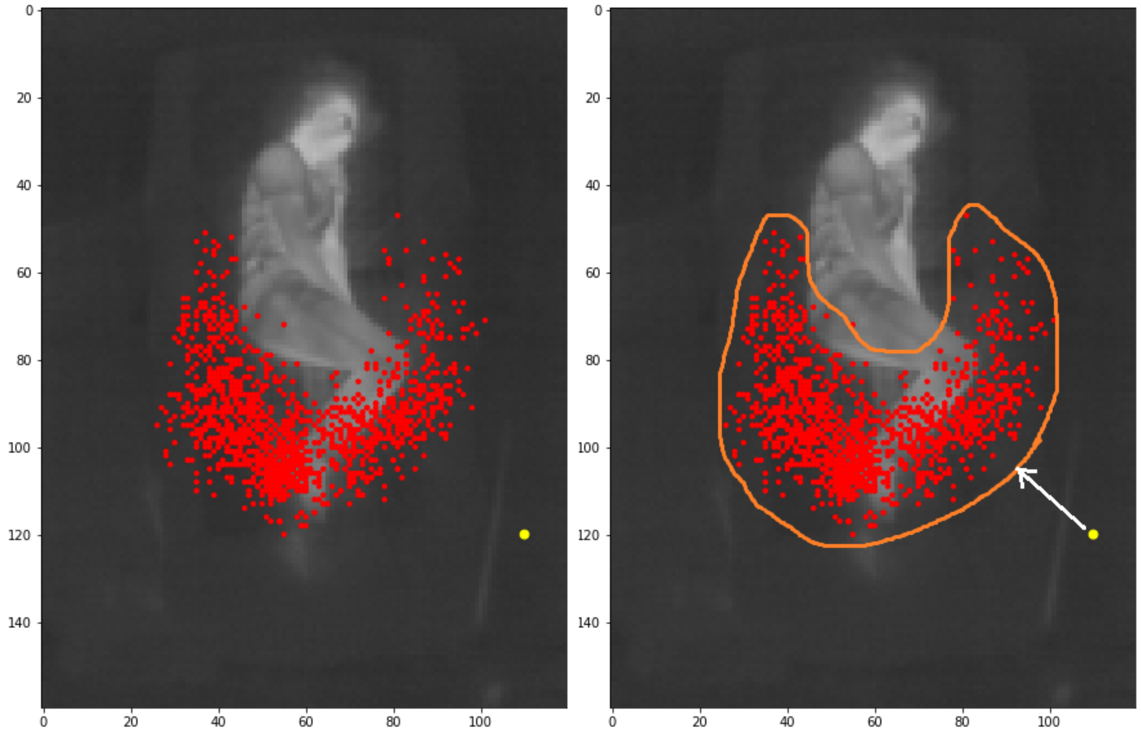
We have applied an ensemble technique which will accumulate predictions from different schemes optimized in different ways and formulate a prediction which shall closely resemble the ground value. Among different schemes, one scheme is noted to be the best found during the validation phase. A circle having radius of a few pixels, λ, is drawn centering the best scheme's prediction. The other predictions being outliers of that circle is eliminated. The remainder predictions are gathered and averaged to obtain the ensemble prediction.




Ensemble
Best Prediction
Circle Around Best Prediction
λ
Eliminate these
Predictions from Various Models
Ground truth
λ Circle Radius
Mean from Existing Points

❖ Post-Processing the Predicted Output

In some cases during our predictions, the estimated key-point of some parts of the human posture faces errors and goes beyond the range of where the actual physical state of the body is. For example, the key point representing the right knee joint may go outside the bed near the edge of the image. As shown in the figure, a prediction for the right knee (depicted in yellow) may go outside the typical range of the right knee joints' points (depicted with red points) gathered from the uncovered labels. Therefore, if that happens, the predicted point is placed just on top of the boundary of the heat-map area created by the red points (depicted in orange) nearest to the prediction. This statistical post-processing is done for each key point and allows the error to go down quite a bit.



❖ Some of our Output Visualized