Flir temperature extraction

Grobel Coralie, Roch David, Chaffard Clement

***Abstract*—Thermal imaging is based upon the study of emitted infrared radiation. The output of thermal camera typically represent an array of the temperature measured at each pixel. In this study, we will use different machine learning models to detect specific parts of the human body, such as the nose and wrists and extract the temperature at these points. With this information we would be able to adjust in real time the air conditioning of a room or building, such that the persons in it neither feel too cold or too warm at any time. For the body parts recognition, we used machine learning algorithms that were trained and rely on RGB images for their prediction. We adapted our temperature array to a grayscale image to make them readable to our models. We compared the different algorithms on RGB and grayscale images to find the best possible model for our study. We found that both image type could give satisfying results when matched with an appropriate model.**

I. Introduction

A. How does thermal imaging work?

Thermal imaging is founded on the study of infrared radiation, which is released by all objects. The amount of radiation emitted is proportional to the overall heat of the object and is detectable in complete darkness.

Thermal pictures are typically grayscale, with white representing heat, black representing cooler regions, and different shades of grey denoting temperature gradients between the two. Newer kinds of thermal imaging cameras, on the other hand, add color to the images they generate in order to assist users in more clearly distinguishing separate objects.

B. Applications of thermal imaging

We will combine thermal imaging with machine learning to measure specific body parts temperatures of multiple persons in a room. The purpose of this study is to be able to adjust in real time the air conditioning of a room or building, such that the persons in it neither feel too cold or too warm at any time.

II. Reproducibility

All our code is available on our github where we detail the libraries we use, the models we use and every parameter we use for each model. The results and plots can be replicated exactly.

III. TEBEL

For this study, we collaborated with TEBEL (Thermal Engineering for the Built Environment Laboratory) . TEBEL focuses on an occupant-centered approach to building energy sufficiency by taking into account people's interior (dis)comfort and well-being through improved design and operation of adaptive thermal systems.

Their research is focused on decreasing building operational energy by stretching the operating temperature of thermal systems via:

- implementing low-temperature emission systems
- personalized indoor thermal conditioning promoting well-being
- smart Machine Learning-based personalized controls

Our project could help them set up an optimal room air conditioning based on persons temperature in real time.

IV. Exploratory Data Analysis

A. A first glimpse at the data

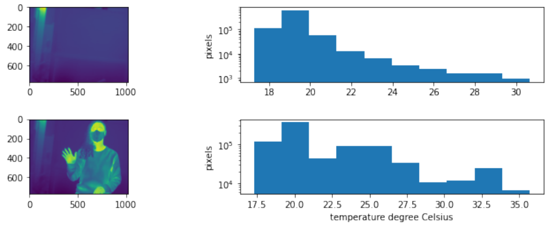


Fig. 1. Thermal image temperature distribution

We can see that a human presence can potentially be deduced from the temperature distribution of the image. We observe a significantly more important fraction of pixels in the range between 21 and 34 degrees.

We can also note that the proportion of pixels is strictly decreasing with temperature increase from 21 degree when there isn't any human body on the image.

However, potential limitations of this idea are numerous. What if the human is far away? Then the variation of the temperature distribution might not be notifiable. What if the surrounding environment already has a temperature similar to the human body, in the 31 to 34 degree Celsius range? Then again we might not notice the presence of the human using the temperature distribution. The question of being able to count how many human bodies are in the picture and location of body parts of interest also emerges.

To be able to face all these challenges, we will use pre-trained human pose recognition algorithms, using Openpifpaf library.

B. How we labeled our data

We now need to label our dataset to be able to compare how well our different models perform. To do so we used an image labeling tool, ImageJ. It allows us to obtain an exploitable csv file containing all coordinates of the body parts of interest. Our body parts of interest were all labeled in the following fashion: Nose, Right eye, Left eye, Front head, Right cheek, Left cheek, Right wrist and Left wrist.

Concerning right and left directions, we set that the labeled right directions correspond to the right of the person in the image. (cf. Fig. 2)

C. Data pre-processing

The different types of images that we use are thermal RGB, grayscale and grayscale normalized. We call thermal RGB images the images mapping the temperature to colors. White is the more warm in the image, then decreasing the temperature goes to yellow, red and blue until the more cold which will be black. To obtain grayscale images from a temperature array, we stretch the temperature range of the array to a 0 to 255 color interval, using the formula with the value min = min(matrix) and max = max(matrix):

$$(matrix - min) * \frac{255}{max - min}$$

This gives us a single channel that is converted to a triple channel (copies of the initial channel) similar to RGB when fed to our models. For the normalized grayscale, the principle is the same than for grayscale, but the min and max values are preset to normalize the images that we send to Openpifpaf. Using the results of the exploratory data analysis, section IV-A, we set min = 20 and max = 40 degrees Celsius. Every pixels resulting to a value below 20 degrees Celsius are set to 0 and every pixels above 40 are set to 255.

V. Models preparation

A. Human pose detection

The machine learning library we used to identify specific body parts in images is Openpifpaf.

It constitutes a generic neural network architecture that detects and constructs a spatio-temporal pose which is composed of a single connected graph whose nodes represent a person's body joints in multiple frames.

Although Openpifpaf wasn't trained on infrared images, we will see that it is still performing accurately when using the types of images defined in section IV-C and apply the appropriate models (cf. section VIII).

B. Implementation of new body parts for Openpifpaf

Some specific parts of the body that we need to have a good estimation of a person temperature were not initially implemented in Openpifpaf. We had to implement 3 new points: forehead, left cheek and right cheek. Our approach to create these new points was purely geometric. The forehead was placed taking the symmetry of the nose with respect to the line which joins the two eyes. In some extreme cases, we obtain a forehead outside the image and therefore the temperature will not be extractable. This didn't cause us any real problem as the point is then ignored. The cheeks were placed taking the vector from the nose to the middle of the eyes and subtracting it to the eyes' position. (cf. Figure 2)

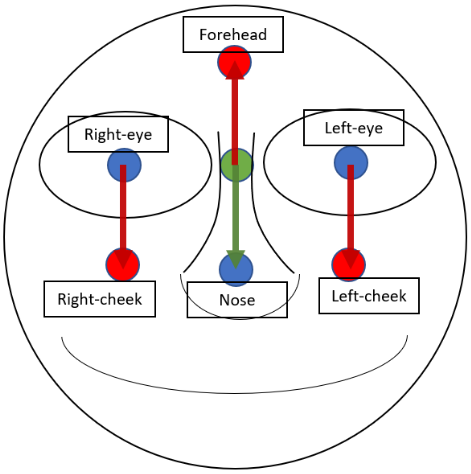


Fig. 2. This represent a face with the different points that we use. The blue dots are the points that are given by Openpifpaf, the green dot is an intermediate point representing the middle of the eyes and the green arrow is the vector from the green dot to the nose. Finally, the three red arrows are ±1 times the green vector, which finally give us the red points representing the part of the face that we were searching for.

VI. Models Evaluation and Discussion

A. Thermal RGB

1) Number of points detected: In order to evaluate the different Openpifpaf models, we first looked at thermal images percentage of detected body parts over all humans. Some models aren't able to detect some points of interest in particular images.

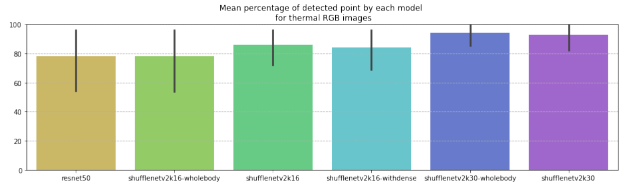


Fig. 3. Mean percentage of body parts detected per model for Thermal RGB images

In figure 3, we can see that most models are able to detect more than 80% of our points of interest. In appendix, table 11, we can see that the points being undetected by the models are in most cases the left and right wrists. Considering only other points, we can see that all model can consistently find them with a probability greater than 94%. We can see that most of the difference between models is due to their ability to reliably detect the left and right wrists. The best performing models, ***shufflenetv2k30-wholebody*** and ***shufflenetv2k30*** are the only models able to detect the wrists more than 70% of the times, when the worst performing model, ***resnet50*** finds the right wrist with only 10% probability. It is important that the models consistently find both eyes and nose as, with our face geometric construction, a missing eye induces a missing cheek on same side as well as a missing forehead, and, a missing nose would also lead to an undetected forehead.

After computing for each model the percentage of points detected over all images, we assessed the precision of the point detected by computing the mean euclidean distance between the predictions of each model and the labels for each body part described in part IV-B.

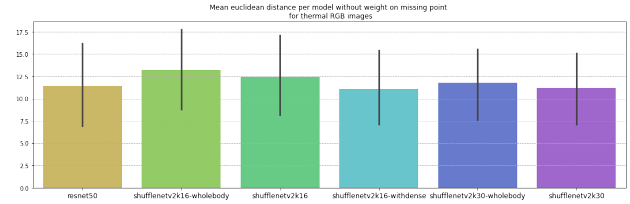


Fig. 4. Mean euclidean distance between predictions and labels of body parts detected per model

2) Euclidean distance to label: We set the error to be the euclidean distance in pixels between the prediction and the corresponding label position.

In figure 4, the error is averaged over all the points that the model detected. It is important to note, that, at this point, a missing point in a model prediction doesn't incur a higher error of the model. We can see in figure 4, that all models share a similar error.

In table 12, we are not surprised to see that the prediction corresponding to the right wrist is the less precise, with an error averaging 20 pixels. The eyes are the point that the model can detect the best with an average error of approximately 4 pixels for both eyes.

Now, we would like to take into account the percentage of points that we found in figure 3 to be able to properly compare our models. To do so, we decided to compute an aggregated loss that will take in consideration both the prediction distance to the label and the percentage of points found by the model.

3) Aggregated loss: But how can we add a penalization term for each missing point to the error when the error represents a distance between a prediction and the corresponding label?

We first thought about adding a constant for each missing point. This constant could then be modified in the case we wanted to penalize more or less a missing point. But we wanted to find a more objective approach. Our second idea was to create a random prediction in the pixel map when a prediction was missing and compute its distance to the corresponding label. But, this would induce a higher loss to missed points in the border of the image compared to missed points in the center. But, we would like our model to more precisely detect points in the center of the image. This is why our final idea was to put the penalization factor equal to the distance between the label and the closest border, when the corresponding prediction was missing. In such a fashion, a point not detected near the border would incur a small loss, which is reasonable as the context available for our models to produce a prediction is smaller. On the other hand, a missed point in the center would lead to maximum penalization, equal to the distance to the closest border.

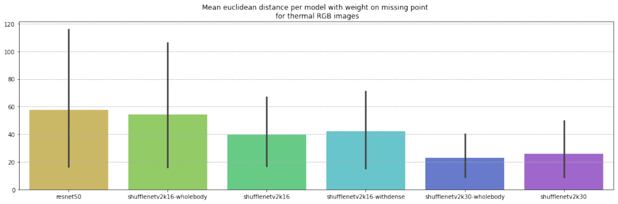


Fig. 5. Aggregated loss of euclidean distance and missing points for Thermal RGB images

In table 13, we can see that, as expected, the models were principally penalized on both wrists. In figure 5, we can identify the best model for thermal RGB images, ***shufflenetv2k30-wholebody***.

VII. Grayscale images

We repeated the same steps on grayscale and normalized grayscale images that we performed on thermal RGB.

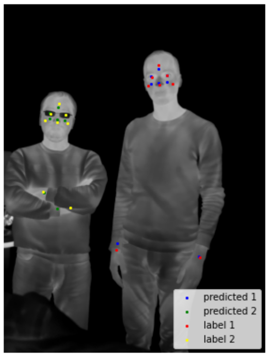


Fig. 6. Image of Openpifpaf prediction on Normalised Gray scale image

Here is an example of the predictions and labels for a normalized grayscale image using ***shufflenetv2k30***.

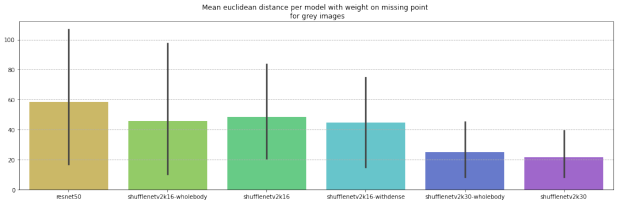


Fig. 7. Aggregated loss of euclidean distance and missing points for grey images

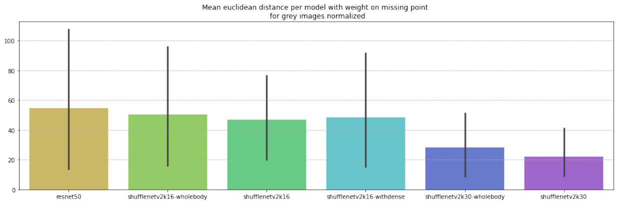


Fig. 8. Aggregated loss of euclidean distance and missing points for normalized grey images

We can confirm looking table 18 that we obtain similar results for our grayscale and normalized grayscale images than for thermal RGB images. The wrists are again poorly detected among the models and responsible for higher losses. But, we obtain as best model, for both grayscale and normalized grayscale, ***shufflenetv2k30***. ***shufflenetv2k30*** differs to ***shufflenetv2k30-wholebody*** by taking a smaller number of keypoints to make its prediction.

False positives and false negatives

To understand more precisely what is happening with our models predictions we computed the false positives (detection of more persons than there are) and false negatives (detection of less persons than there are). We found that a common pattern was the separation of a person into multiple persons by the models. The ratio of false positive was averagely around 20% when the ratio of false negative was rather around 5%. (cf Table 31)

VIII. Comparing best models on RGB and grayscale images

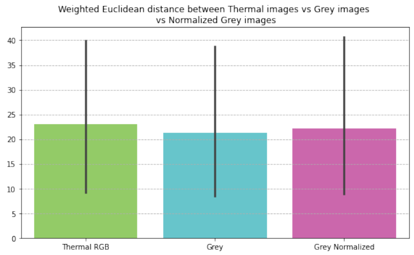


Fig. 9. Aggregated loss of euclidean distance and missing points between image types

We compare the results for thermal RGB, grayscale and normalized grayscale in figure 9. We observe a very similar aggregated error for every type of image. The very large 95% confidence interval is probably due to the small amount of data we had at disposition. Although we can't establish on which image type the respective best model works better, we can affirm that specific body parts recognition works very similarly on thermal RGB images and grayscale images.

IX. To go further

A. *Time and streaming*

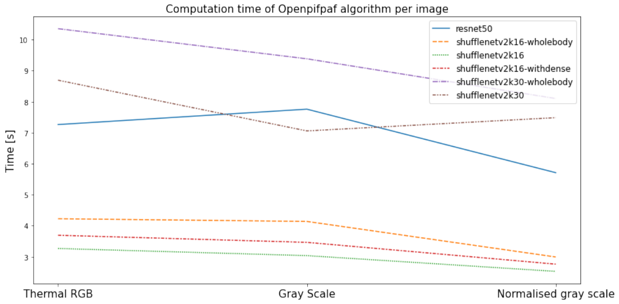


Fig. 10. Time per model per type of images

In figure 10 we see that our best models, ***shufflenetv2k30-wholebody*** and ***shufflenetv2k30*** also constitute the slowest models being able to treat an image every 9 seconds in average. ***shufflenetv2k16*** works much faster and takes only 3 seconds to handle an image, making it the ideal model for body parts recognition while streaming. We can also note that all models tend to perform slightly faster on normalized grayscale images. This could be due to the fact that these images contain less information.

B. *Real time camera access (Fribourg)*

In the best possible scenario, we would like to have a direct stream from the camera to our algorithm. In those conditions, we will be able to have the temperature of different body parts of the people in the image in ”live” (only with a phase shift due to computation time). You can find the file using the webcam to do it in the folder ”code”. Follow the instructions in the readme to execute it.

To achieve this goal, the most relevant way we tried were to use the spinnaker-sdk and the simple-pyspin library. Unfortunately, those weren't compatible with the camera that we had. After discussion with the flir helpdesk, we conclude that we should use the atlas-sdk. To be able to install ***atlas-sdk*** it was required to complete a form and be accepted by flir. Due to the short amount of time remaining for our project and the camera being located in Fribourg, we couldn't achieve this goal.

X. Conclusion

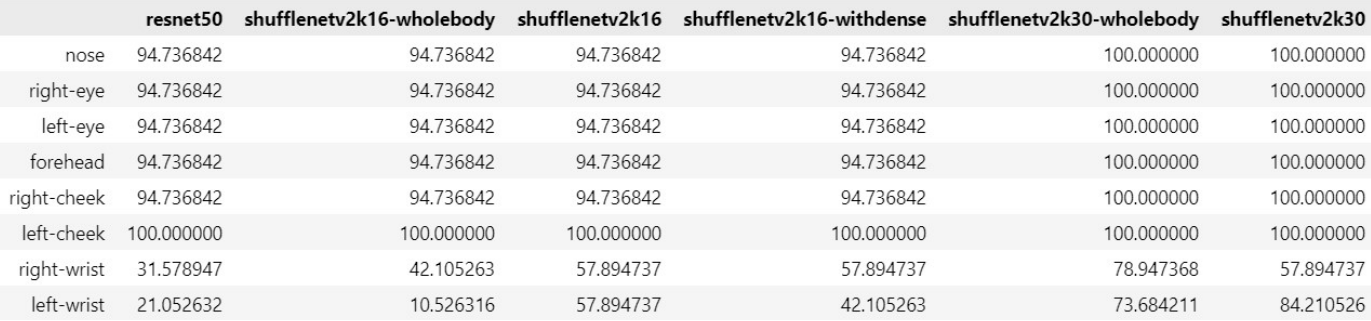
We were able to correctly locate specific body parts with very satisfying precision and extract temperature on both RGB and grayscale image. Thermal RGB images should be used together with ***shufflenetv2k30-wholebody*** wheras grayscale images gives better results with ***shufflenetv2k30*** model from Openpifpaf. Our program is able to store for each image passed in argument the temperature found for the required body parts, main limitations being Openpipaf models prediction accuracy as well as the camera temperature precision. Our body parts temperature extraction program has also been implemented to work on live streams.

References

[1] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. “Pifpaf: Composite fields for human pose estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11977–11986.

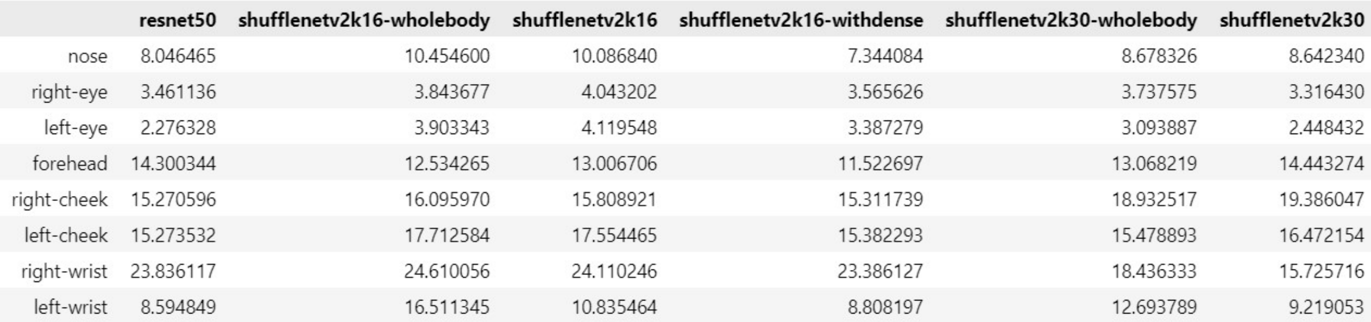
[2] J Michael Lloyd. *Thermal imaging systems*. Springer Science & Business Media, 2013.

APPENDIX



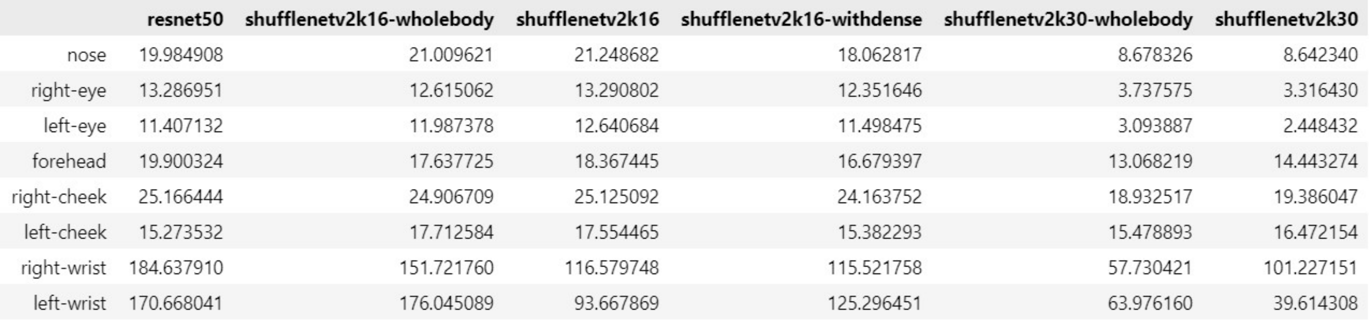
	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	94.736842	94.736842	94.736842	94.736842	100.000000	100.000000
right-eye	94.736842	94.736842	94.736842	94.736842	100.000000	100.000000
left-eye	94.736842	94.736842	94.736842	94.736842	100.000000	100.000000
forehead	94.736842	94.736842	94.736842	94.736842	100.000000	100.000000
right-cheek	94.736842	94.736842	94.736842	94.736842	100.000000	100.000000
left-cheek	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
right-wrist	31.578947	42.105263	57.894737	57.894737	78.947368	57.894737
left-wrist	21.052632	10.526316	57.894737	42.105263	73.684211	84.210526

Fig. 11. Mean percentage of body parts detected per model for thermal images



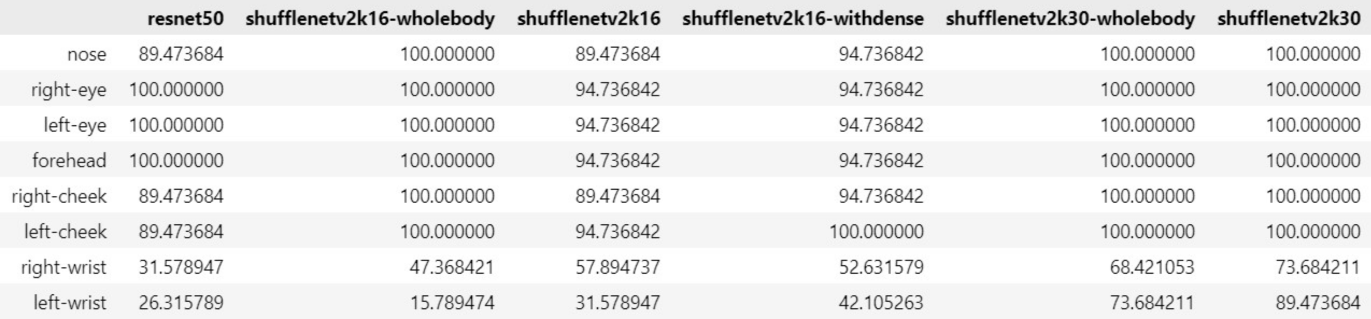
	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	8.046465	10.454600	10.086840	7.344084	8.678326	8.642340
right-eye	3.461136	3.843677	4.043202	3.565626	3.737575	3.316430
left-eye	2.276328	3.903343	4.119548	3.387279	3.093887	2.448432
forehead	14.300344	12.534265	13.006706	11.522697	13.068219	14.443274
right-cheek	15.270596	16.095970	15.808921	15.311739	18.932517	19.386047
left-cheek	15.273532	17.712584	17.554465	15.382293	15.478893	16.472154
right-wrist	23.836117	24.610056	24.110246	23.386127	18.436333	15.725716
left-wrist	8.594849	16.511345	10.835464	8.808197	12.693789	9.219053

Fig. 12. Euclidean distance for thermal images



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	19.984908	21.009621	21.248682	18.062817	8.678326	8.642340
right-eye	13.286951	12.615062	13.290802	12.351646	3.737575	3.316430
left-eye	11.407132	11.987378	12.640684	11.498475	3.093887	2.448432
forehead	19.900324	17.637725	18.367445	16.679397	13.068219	14.443274
right-cheek	25.166444	24.906709	25.125092	24.163752	18.932517	19.386047
left-cheek	15.273532	17.712584	17.554465	15.382293	15.478893	16.472154
right-wrist	184.637910	151.721760	116.579748	115.521758	57.730421	101.227151
left-wrist	170.668041	176.045089	93.667869	125.296451	63.976160	39.614308

Fig. 13. Aggregated loss of euclidean distance and missing points for thermal images



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	89.473684	100.000000	89.473684	94.736842	100.000000	100.000000
right-eye	100.000000	100.000000	94.736842	94.736842	100.000000	100.000000
left-eye	100.000000	100.000000	94.736842	94.736842	100.000000	100.000000
forehead	100.000000	100.000000	94.736842	94.736842	100.000000	100.000000
right-cheek	89.473684	100.000000	89.473684	94.736842	100.000000	100.000000
left-cheek	89.473684	100.000000	94.736842	100.000000	100.000000	100.000000
right-wrist	31.578947	47.368421	57.894737	52.631579	68.421053	73.684211
left-wrist	26.315789	15.789474	31.578947	42.105263	73.684211	89.473684

Fig. 14. Mean percentage of body parts detected per model for grey images

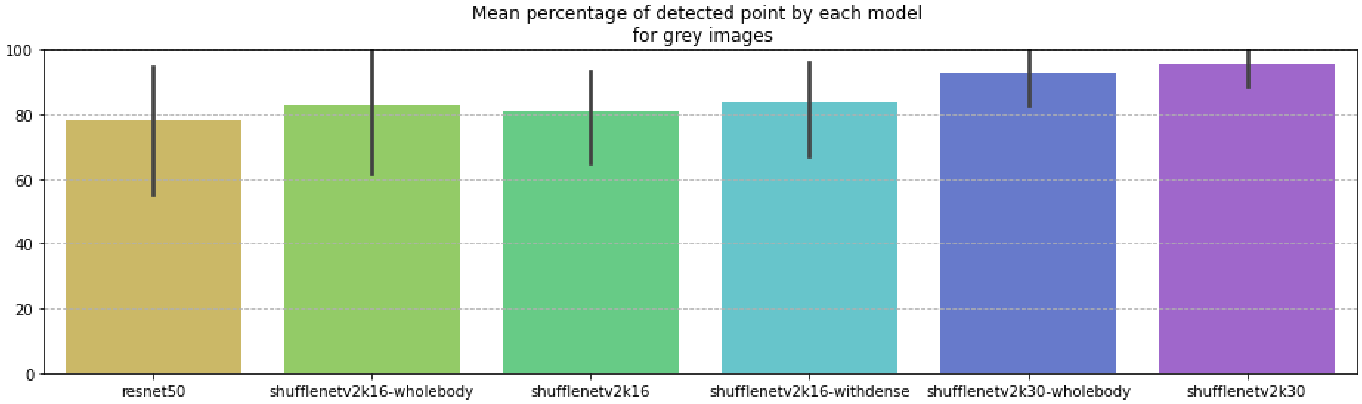
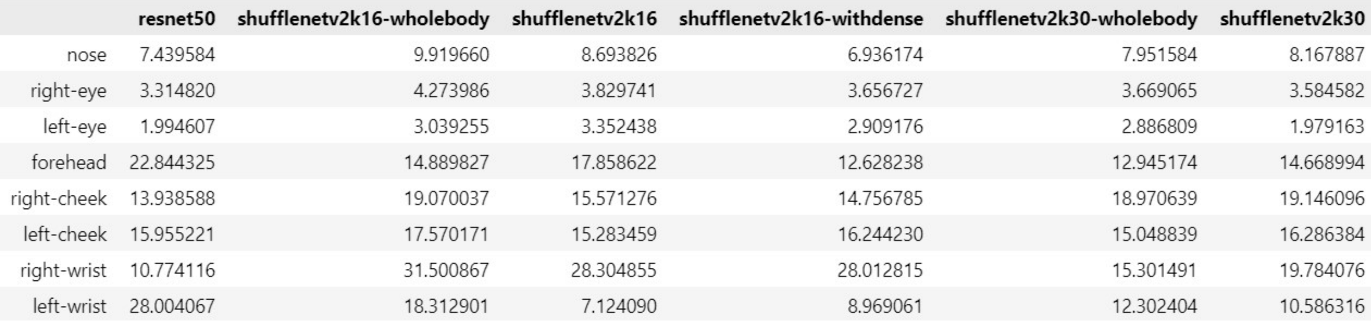


Fig. 15. Mean percentage of body parts detected per model for grey images



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	7.439584	9.919660	8.693826	6.936174	7.951584	8.167887
right-eye	3.314820	4.273986	3.829741	3.656727	3.669065	3.584582
left-eye	1.994607	3.039255	3.352438	2.909176	2.886809	1.979163
forehead	22.844325	14.889827	17.858622	12.628238	12.945174	14.668994
right-cheek	13.938588	19.070037	15.571276	14.756785	18.970639	19.146096
left-cheek	15.955221	17.570171	15.283459	16.244230	15.048839	16.286384
right-wrist	10.774116	31.500867	28.304855	28.012815	15.301491	19.784076
left-wrist	28.004067	18.312901	7.124090	8.969061	12.302404	10.586316

Fig. 16. Euclidean distance for grey images

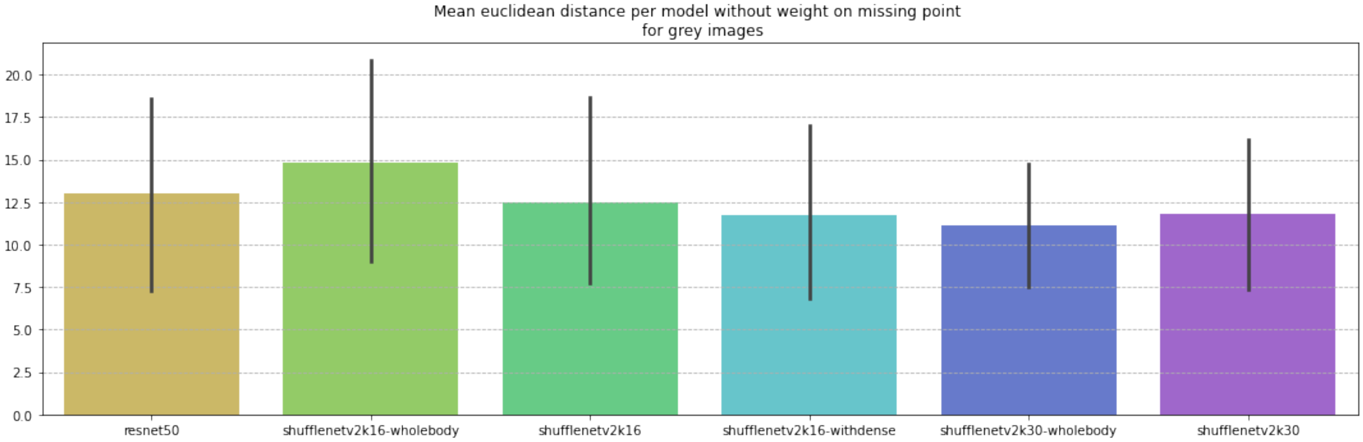
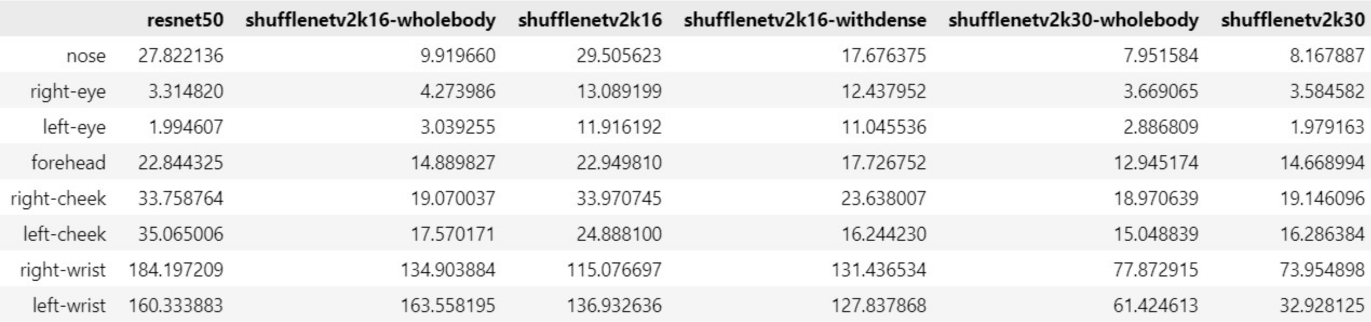


Fig. 17. Euclidean distance for grey images



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	27.822136	9.919660	29.505623	17.676375	7.951584	8.167887
right-eye	3.314820	4.273986	13.089199	12.437952	3.669065	3.584582
left-eye	1.994607	3.039255	11.916192	11.045536	2.886809	1.979163
forehead	22.844325	14.889827	22.949810	17.726752	12.945174	14.668994
right-cheek	33.758764	19.070037	33.970745	23.638007	18.970639	19.146096
left-cheek	35.065006	17.570171	24.888100	16.244230	15.048839	16.286384
right-wrist	184.197209	134.903884	115.076697	131.436534	77.872915	73.954898
left-wrist	160.333883	163.558195	136.932636	127.837868	61.424613	32.928125

Fig. 18. Aggregated loss of euclidean distance and missing points for grey images

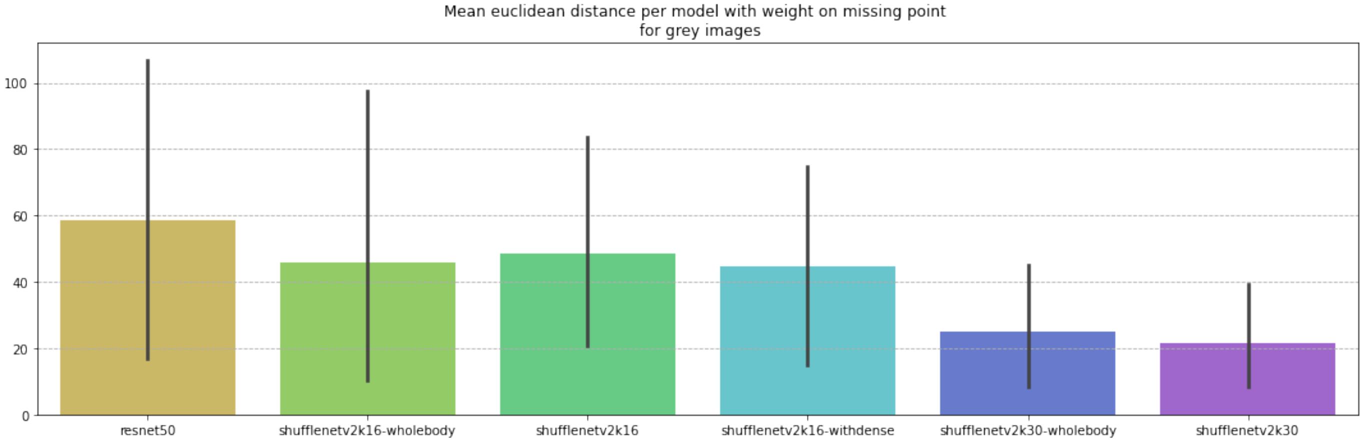
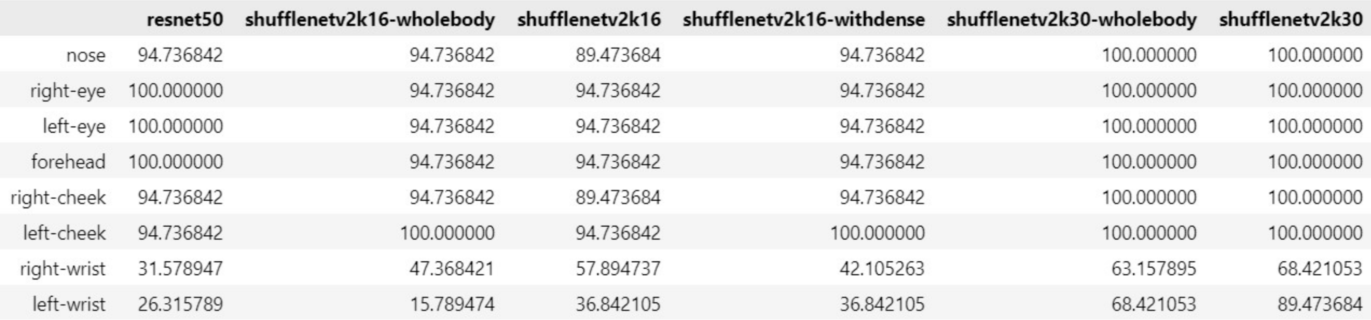


Fig. 19. Aggregated loss of euclidean distance and missing points for normalized grey images



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	94.736842	94.736842	89.473684	94.736842	100.000000	100.000000
right-eye	100.000000	94.736842	94.736842	94.736842	100.000000	100.000000
left-eye	100.000000	94.736842	94.736842	94.736842	100.000000	100.000000
forehead	100.000000	94.736842	94.736842	94.736842	100.000000	100.000000
right-cheek	94.736842	94.736842	89.473684	94.736842	100.000000	100.000000
left-cheek	94.736842	100.000000	94.736842	100.000000	100.000000	100.000000
right-wrist	31.578947	47.368421	57.894737	42.105263	63.157895	68.421053
left-wrist	26.315789	15.789474	36.842105	36.842105	68.421053	89.473684

Fig. 20. Mean percentage of body parts detected per model for normalized grey images

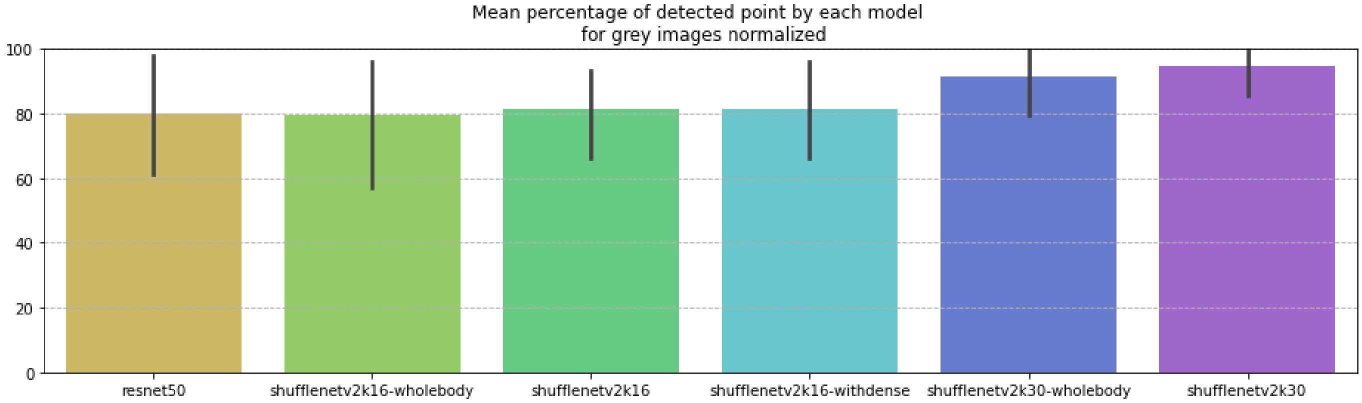
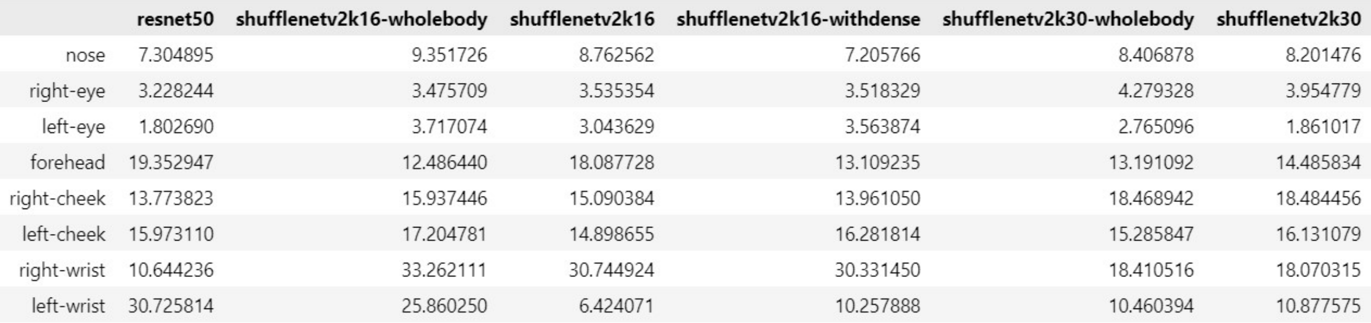


Fig. 21. Mean percentage of body parts detected per model for grey images



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	7.304895	9.351726	8.762562	7.205766	8.406878	8.201476
right-eye	3.228244	3.475709	3.535354	3.518329	4.279328	3.954779
left-eye	1.802690	3.717074	3.043629	3.563874	2.765096	1.861017
forehead	19.352947	12.486440	18.087728	13.109235	13.191092	14.485834
right-cheek	13.773823	15.937446	15.090384	13.961050	18.468942	18.484456
left-cheek	15.973110	17.204781	14.898655	16.281814	15.285847	16.131079
right-wrist	10.644236	33.262111	30.744924	30.331450	18.410516	18.070315
left-wrist	30.725814	25.860250	6.424071	10.257888	10.460394	10.877575

Fig. 22. Euclidean distance for normalized grey images

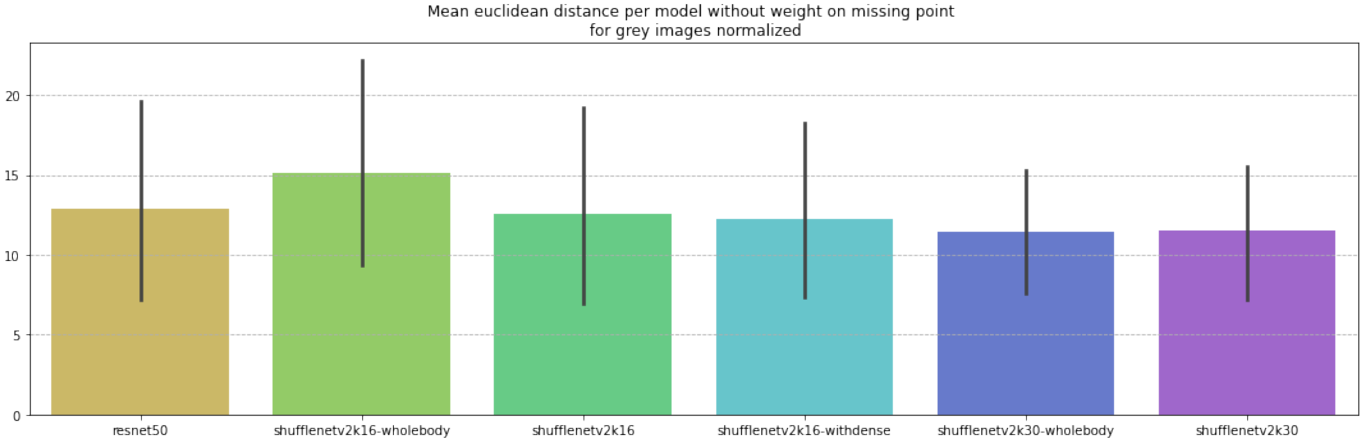
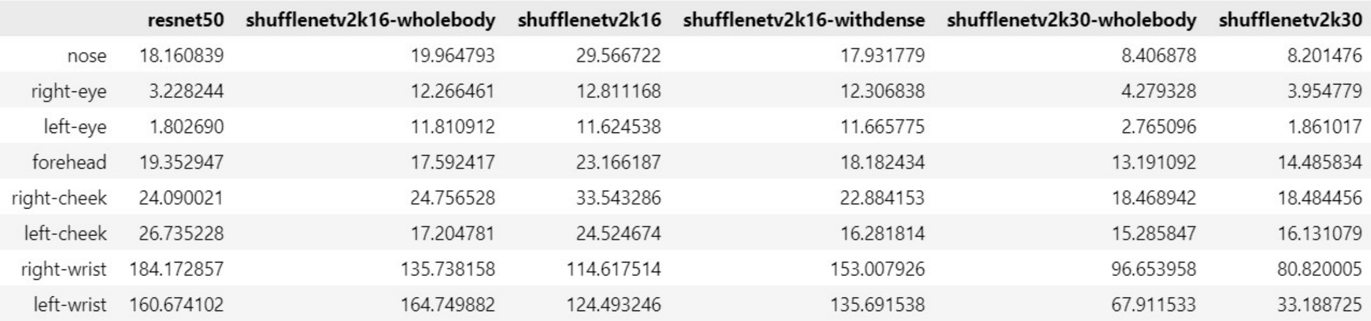


Fig. 23. Euclidean distance for normalized grey images



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
nose	18.160839	19.964793	29.566722	17.931779	8.406878	8.201476
right-eye	3.228244	12.266461	12.811168	12.306838	4.279328	3.954779
left-eye	1.802690	11.810912	11.624538	11.665775	2.765096	1.861017
forehead	19.352947	17.592417	23.166187	18.182434	13.191092	14.485834
right-cheek	24.090021	24.756528	33.543286	22.884153	18.468942	18.484456
left-cheek	26.735228	17.204781	24.524674	16.281814	15.285847	16.131079
right-wrist	184.172857	135.738158	114.617514	153.007926	96.653958	80.820005
left-wrist	160.674102	164.749882	124.493246	135.691538	67.911533	33.188725

Fig. 24. Aggregated loss of euclidean distance and missing points for normalized grey images

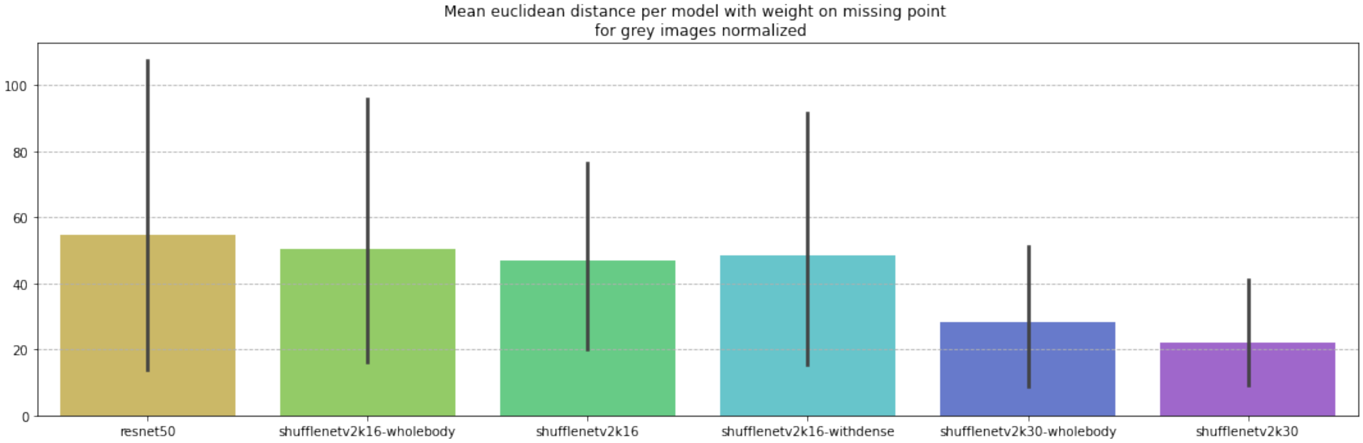
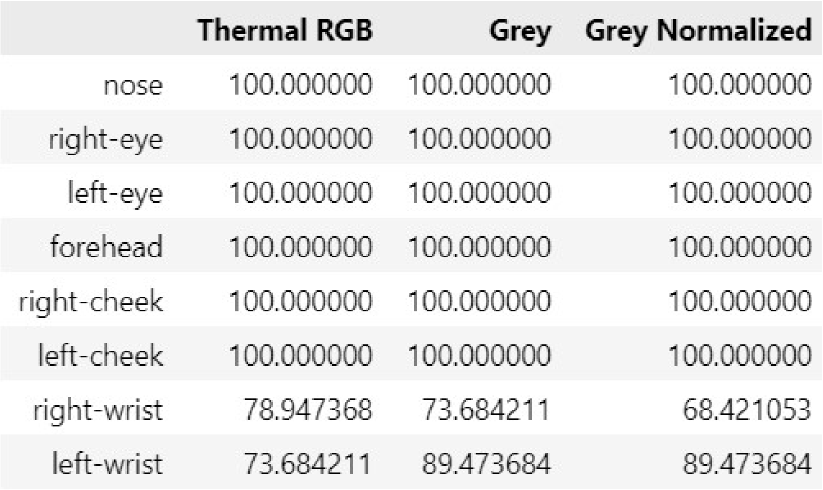


Fig. 25. Aggregated loss of euclidean distance and missing points for normalized grey images



	Thermal RGB	Grey	Grey Normalized
nose	100.000000	100.000000	100.000000
right-eye	100.000000	100.000000	100.000000
left-eye	100.000000	100.000000	100.000000
forehead	100.000000	100.000000	100.000000
right-cheek	100.000000	100.000000	100.000000
left-cheek	100.000000	100.000000	100.000000
right-wrist	78.947368	73.684211	68.421053
left-wrist	73.684211	89.473684	89.473684

Fig. 26. Difference of percentage of finding point between image types

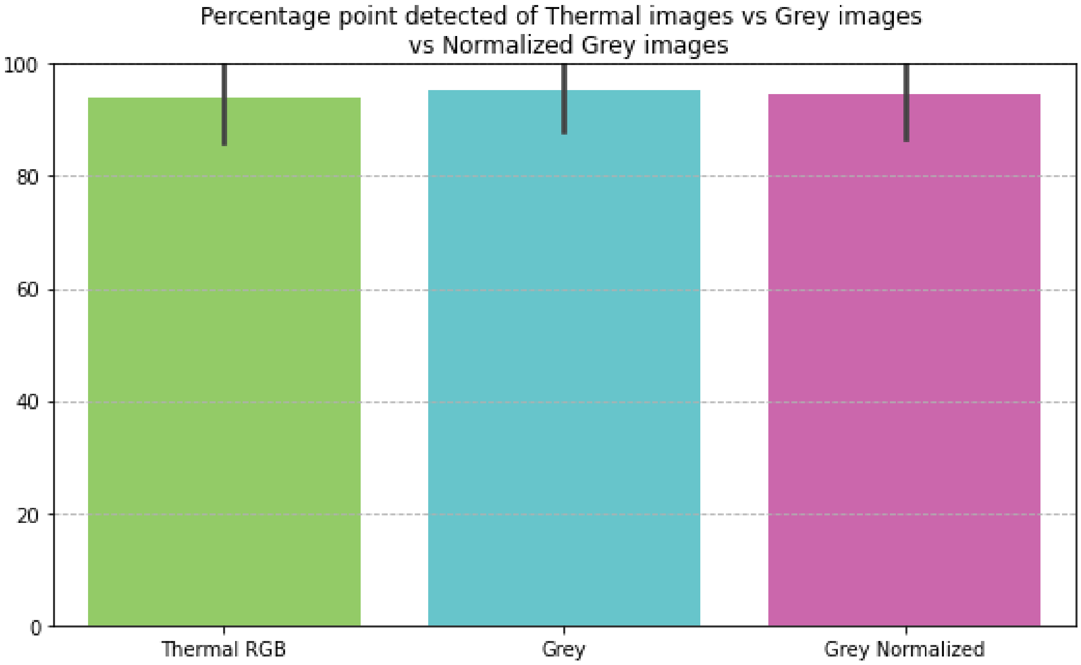
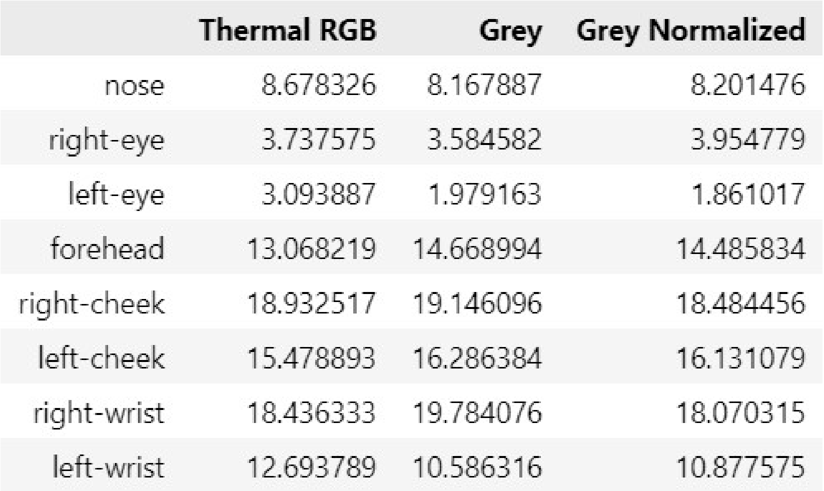


Fig. 27. Mean difference of percentage of finding point between image types



	Thermal RGB	Grey	Grey Normalized
nose	8.678326	8.167887	8.201476
right-eye	3.737575	3.584582	3.954779
left-eye	3.093887	1.979163	1.861017
forehead	13.068219	14.668994	14.485834
right-cheek	18.932517	19.146096	18.484456
left-cheek	15.478893	16.286384	16.131079
right-wrist	18.436333	19.784076	18.070315
left-wrist	12.693789	10.586316	10.877575

Fig. 28. Euclidean distance difference between image types

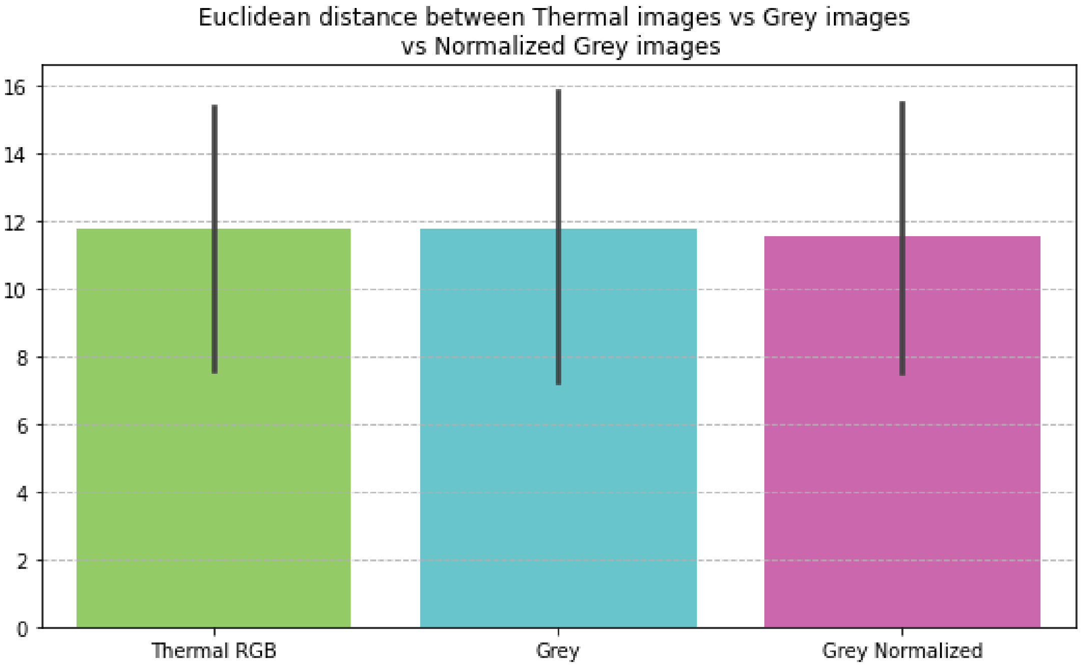
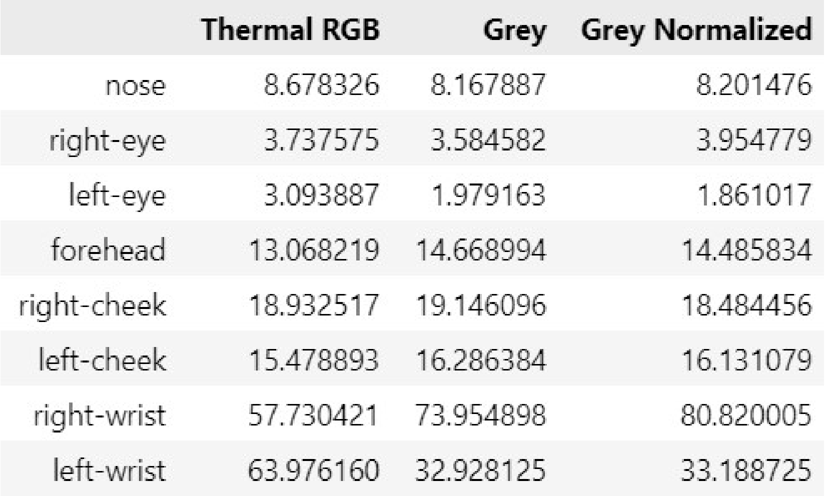
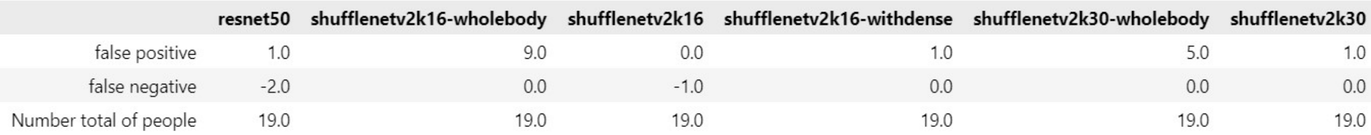


Fig. 29. Mean difference of percentage of finding point between image types



	Thermal RGB	Grey	Grey Normalized
nose	8.678326	8.167887	8.201476
right-eye	3.737575	3.584582	3.954779
left-eye	3.093887	1.979163	1.861017
forehead	13.068219	14.668994	14.485834
right-cheek	18.932517	19.146096	18.484456
left-cheek	15.478893	16.286384	16.131079
right-wrist	57.730421	73.954898	80.820005
left-wrist	63.976160	32.928125	33.188725

Fig. 30. Weighted euclidean distance difference between image types



	resnet50	shufflenetv2k16-wholebody	shufflenetv2k16	shufflenetv2k16-withdense	shufflenetv2k30-wholebody	shufflenetv2k30
false positive	1.0	9.0	0.0	1.0	5.0	1.0
false negative	-2.0	0.0	-1.0	0.0	0.0	0.0
Number total of people	19.0	19.0	19.0	19.0	19.0	19.0

Fig. 31. People that have not been detected or people that doesn't exist and have been detected on Thermal RGB images