Long-term Tracking of Human Skin Areas on Thermal Images for Monitoring Vital Signs

Zak Morgan, Byoung Hun Min, Seunghoi Kim, Mahdi Nasrollahi
University College London, Department of Computer Science
Malet Place Engineering Building, Gower Street, London WC1E 6BT
Email: zak.morgan.17@ucl.ac.uk, brian.min.15@ucl.ac.uk, kim.kim.17@ucl.ac.uk, mahdi.nasrollahi.17@ucl.ac.uk

Abstract—**Object tracking in videos is a widely studied problem, and we focused specifically on tracking areas of human skin in the thermal domain. This is an interesting area since its possible to get insights into physiological signals such as breathing, pulse rate and even emotions since it influences facial temperatures. It also has some unique properties not shared by RGB images, such as the ability to "see into the past" by observing heat traces left behind on surfaces. This can all be done in pitch black as-well without requiring light due to the nature of thermal imaging. Since thermal tracking is much less studied than RGB tracking, and has much less data available we have proposed turning thermal videos into synthetic RGB videos using a conditional generative adversarial network (cGAN), specifically a pix2pix-based architecture. We then applied state of the art RGB tracking algorithms to this. We found an increase in accuracy and robustness in best-case clips similar to our training data, and also crucially when tracking areas harder to see in thermal images, such as the mouth and nostrils. We have also released our dataset, and a number of tools to make collection of similar datasets easier.**

1. Introduction

Long-term tracking is a process wherein a bounding box is specified containing our target in an initial frame, and then drawn around it in all subsequent frames if present, displaying a signal indicating absence otherwise. Where this differs from normal short term tracking is the indicator for absence, handling of re-detection upon the object re-entering the frame, and also running for an indefinitely long period of time [1]. Tracking of objects in thermal imagery and RGB imagery pose different problems, where RGB trackers are optimized based on contours, colours and are usually fairly high resolution, thermal images mostly possess low resolution, less defined edges (Less high-frequency details) and are single channel. Hence it would seem algorithms created for RGB tracking are not suitable to apply to thermal tracking. Moreover, thermal images have dynamic temperture characteristics. They are affected by both internal and external influences such as breathing, physiological effects, and external environmental factors.

This is supported by the evidence in the results of the VOT2017-TIR and VOT2017 benchmarks, where algorithms that come top in RGB tracking perform badly in TIR(Thermal Infra-Red) tracking and vice versa. For example SRDCFir (Spatially Regularized Discriminative Correlation Filters) comes 3rd in TIR tracking, but SRDCF [2] (its RGB counterpart) comes 40th in RGB tracking [3]. We see a similar story for MOSSEca [4](Minimum Output Sum of Squared Error context aware) placing 4th, and 34th respectively. The one exception appears to be ECO [5](Efficient Convolution Operators), placing 5th in TIR, and 4th in RGB, however 5th is much less impressive considering it is out of 10 and only scores about 50% of the EAO (Expected Average Overlap) of the top DSLT tracker [3].

RGB tracking is far more robust and accurate than thermal tracking [6], partly due to the larger amount of data [7] available to train the common CNN-based architectures we see dominating today.

From this the natural question to ask is if we have better results in RGB tracking, what do thermal images give us that ones in the visual wavelength of light don't? Thermal images are not affected by lighting, meaning we can use them to track objects 24/7 including in pitch black. They also are seen as more private, since if you look at a thermal image, you can't instantly tell who it portrays, mostly thanks to the low-resolution often exhibited by these sensors. This is actually wrong though and its even thought that thermal facial recognition could be more accurate than that in normal RGB images [7]. It is however true that since there is no large dataset of everyone's thermal facial signature it is harder currently, also helped by the fact that you would have to account for variation in wind, ambient temperature and the metabolic processes of the subject. This combined with the large cost of thermal sensors means that currently, it is slightly less invasive than visible images since we cannot match them to a known image.

In this paper we therefore have proposed a thermal tracking algorithm which converts TIR sequences into synthetic RGB sequences before applying state of the art RGB tracking algorithms to it. This solution showed promise due to previous works that show it can work doing the reverse, that is generating synthetic TIR data to train an algorithm for TIR tracking [8]. We evaluate and discuss our results finding that depending on the benchmark and RGB tracking algorithm used we can indeed improve tracking using our method. Also discussed is the possibility of using facial landmarking algorithms on the synthetic RGB sequences produced, which is not possible on the TIR sequences. Finally we also present a streamlined method to collect training and testing data using the FLIRONE camera, as-well as the dataset we collected.

2. Related Work

The key piece of related work is Zhang, L *et al.* [8] which showed state of the art tracking results on thermal images by training on a synthetic data set. This synthetic dataset was made using a Conditional Generative Adversarial Network (cGAN), pix2pix, designed by Isola *et al.* [9] trained on pairs of rgb and thermal images to translate rgb images into synthetic thermal images. Given that algorithms in the visual domain are much more advanced than in the thermal domain [6], we propose converting thermal images to synthetic rgb to then apply standard state of the art trackers.

Another approach seen in [8] is using unlabelled training data, using for example cycleGAN [10]. This was found to produce much less accurate results and so should only be used if its a massive barrier to get labelled training data.

It is important to look at how state of the art RGB trackers operate, if used on synthetic RGB results. The state of the art as of the VOT2018-LT tracker results was MBMD [11]. It is comprised of 2 parts R a fixed regression network and a verification network V architecturally similar to MDNet [12] which was the best performing algorithm in VOT-2015. MDNet is a multi-domain CNN (Convolutional Neural Net) meaning we have different non-shared branches for each domain and a number of shared layers which are kept after training on each domain. These shared layers therefore ideally will then form a generalized tracker.

The added R over just the V makes MBMD an improvement over MDNet since it partially solves the problem of updating with bad samples. This reduces drift since samples are assessed by both V and R, V is less likely to be impacted by noisy or inaccurate updates.

Its important to also look at the state of the art for thermal tracking, since that is what we are trying to improve upon. These include such algorithms as DSLT [13], EBT [14] and SRDCFir [2]. It should be noted all three of these were initially designed for RGB tracking, with EBT coming 3rd and SRDCF coming 4th in the VOT-2015 challenge [15]. DSLT was not submitted but performed better than all entrants in the 2014 version of the challenge. This again shows that thermal tracking is behind RGB tracking since these algorithms now place on the order of 60th place in the 2018 challenge [16] performing as much as 4 times worse than the leading algorithms, while they are still the top performing algorithms in TIR tracking [3].

Of note is that these three trackers are all of different types, SRDCFir is a DCF (Discriminative Correlation Filter) tracker, EBT is a tracking by detection based algorithm and DSLT is deep regression tracker. This shows there is not a clear winner in terms of approach to this problem yet, unlike RGB tracking where DCF and CNN methods account for the majority of trackers [16].

3. Proposed Solution

We propose that performing tracking on a synthetic RGB video, generated from a raw thermal matrix, will perform better than tracking directly on the raw thermal matrix. The reason for believing this is that older state of the art tracking algorithms such as TLD, KCF and MIL which perform accurately and robustly in the RGB domain, do not perform so well in the thermal domain [17]. More modern algorithms are often based around Convolution Neural Nets (CNN) which require extensive training, which we lack the data to train in the thermal domain [8]. Since we have established that state of the art RGB trackers are more advanced than thermal trackers, we leveraged that to track in thermal images.

Specifically we generated these synthetic RGB videos from a thermal matrix using an architecture similar to pix2pix [9]. We have added some layers to use higher resolution images to preserve more details in our thermal image. This is because if we want to track smaller features such as a nostril, we don't want to remove these when downsampling the image for input into the network.

4. Experiment

4.1. Methodology

Our experiment is comprised of a few steps, data collection, training, benchmarking and then finally comparison with existing solutions. To train pix2pix we needed a large collection of paired data, that is matching thermal and RGB images. We are using pix2pix since in previous research such as Zhang, L et al. showed paired data generation using pix2pix to be superior to other options such as CycleGAN which train on un-paired data [8].

Benchmark videos are then fed one frame at a time into the trained network to turn them into synthetic RGB videos. Trackers can now be benchmarked on both the new processed video and the original. We used the implemented tracking algorithms in OpenCV2 to test if we can get a better result on the synthetic RGB video than the original thermal video.

What we change is therefore our video input, changing between thermal videos and RGB videos. What we measure will be our evaluation metrics including Accuracy (Average Overlap) and Robustness which are based on the Visual Object Tracking benchmark [18]. To make sure we get accurate results we will of course be using the same videos for tracking in both RGB and thermal, as well as making sure we choose a suitably difficult clip to track on.

4.2. Dataset Description and Collection

To track a face we therefore wanted a facial data set to match against, however there is limited data online and much of it is either not of faces such as the FLIR car dataset [19], or there was sufficient parallax error in the images such that the thermal and RGB pairs could not be registered properly such as the OTCBVS Dataset 02 [20]. These also provided sub-optimal lighting conditions for the RGB images, thus meaning if we train on them we will also get badly lit synthetic RGB images which are hard to track.

To collect data, a FLIR ONE Pro camera was used, attached to an android phone. We only collected images of ourselves due to ethical reasons. Since we are converting from thermal to RGB for the purpose of tracking, we want the best possible RGB image outputted. Thus we made sure our lighting was ideal and then took photos in a variety of ambient temperatures and with varying facial temperatures too for example, having a cold nose.

Now the images must be processed, first separating the thermal matrix and RGB image from the singular JPEG file that the flir camera produces. We do this using the “Flir Image Extractor” python library [21]. We then applied “optimal quantization” to our thermal matrix, a process outlined in Cho, Youngjun *et al.* [22] and implemented in the TIPA python library [23]. This quantization first removes thermal signals beyond 1.96 standard deviations and then iteratively analyses the histogram of the thermal matrix to find the best separation between the background and the human face contained within the image. This is done because in long term tracking the temperature of the face overall can change, so we change the range of temperatures we look at every frame. In standard RGB this would be like changing our exposure every frame to account for changes in illumination.

Now this is done a quick scale and crop of the RGB image to register it to match the thermal image is performed. Then they are put side by side in one image file since this makes pairing the images in the training code easier. Figure 1 shows the path the image follows through the pipeline.

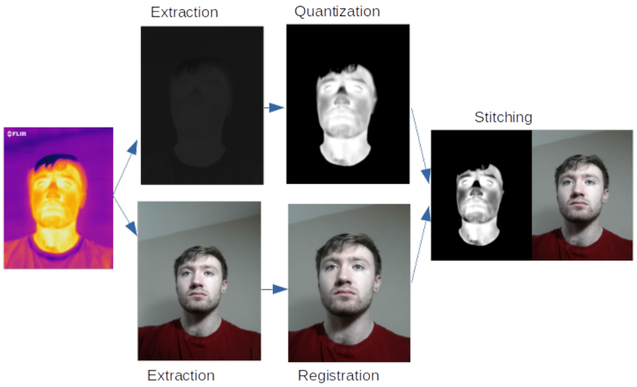

Figure 1. Post Processing Data Collection Pipeline

The final step is to separate the dataset into training and validation data, with some optional test images thrown in too. There is no set way to do this, although there have been some attempts to construct rules such as by Guyon, I [24]. We followed a split commonly seen that is about 75% training and 25% validation. This is because of our somewhat small data-set, if we had more then we would shift towards a 90/10 split as also often seen. This dataset ended up being comprised of 299 images from 6 participants. When splitting this data we tried to make sure we had one of each situation in both the testing and validation set. For example if we had a background in the validation set, that wasn’t present in the training set our model would struggle to learn.

Our testing data was collected a little differently due to the different format. To collect raw thermal matrix videos we made a small android application using the FLIR mobile SDK. Due to there being no API available to get the raw thermal matrix, we instead had to create a measurement for every pixel, and then collect those measurements. We then applied optimal quantization to this. We also ran it through our neural net, to achieve a real thermal video, as-well as the synthetic RGB video to compare tracking on. We ended up recording only one of these testing videos, with one subject who was also used in the training data. The other testing videos came from pre-existing datasets.

4.3. Evaluation Metrics

Our evaluation metrics are based on the VOT2019 [25] performance evaluation protocol which includes three primary measures: accuracy, robustness and expected average overlap (EAO).
Accuracy is measured by calculating the average overlap between the predicted and ground truth bounding boxes. For each frame, the Intersection over Union (IoU) value is calculated and averaged over all successfully tracked frames which results in a value between 0 and 1, where 1 represents a complete overlap. Robustness simply measures the number of times the tracker loses the target. This value is then normalised between 0 and 1.
EAO is an estimator of the average overlap a tracker is expected to attain on a large set of sequences with the same visual properties as the given dataset. This extends the Accuracy metric by addressing the issue of increased variance and bias of average overlap due to varying lengths of sequences. However, we did not compute this metric as our sequences have consistent lengths of sequences. By measuring the accuracy and robustness of each tracker, we analysed the performance of some state-of-the-art trackers on facial thermal datasets and compared their performances to when the same trackers were applied to the same facial datasets converted into an RGB format.

4.4. Benchmarks

Due to the lack of existing benchmarks for facial TIR dataset, we manually labelled the ground-truth bounding boxes of available facial thermal data using an open source tool available on GitHub called Visual Object Tagging Tool (VOTT) [26] by Microsoft. We were able to collect five sets of facial thermal sequences of varying difficulty in terms of tracking. Our benchmark consists of these five sets labelled as such: (1) Slow Facial Movements, (2) Facial Movements with Partial Occlusion, (3) Rapid Facial Movements, (4) Minimal Facial Movements (Mouth), and (5) Minimal Facial Movements (Nostril). These are presented in table 1.

#	Dataset	Description	Frames	ROI
1	P104AQ6	Slow Facial Movements	200	Nose
2	P104AQ8	Partial Occlusion	200	Nose
3	P109AQ8	Rapid Facial Movements	200	Nose
4	Zak's Face	Moderate Facial Movements	74	Mouth
5	Zak's Face	Moderate Facial Movements	74	Nostril

TABLE 1. Benchmark Facial Dataset Information

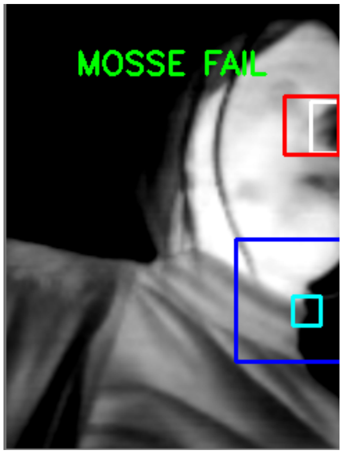

Figure 2. Raw Thermal Sequence - Frame #180 from Sequence # 3

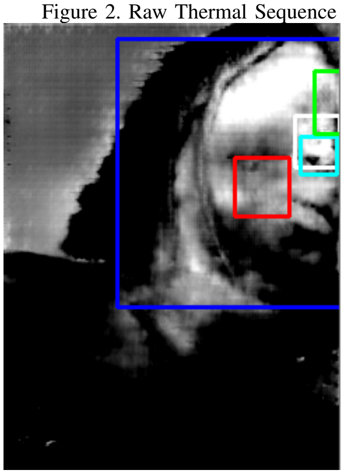

Figure 3. Generated RGB Sequence - Frame #200 from Sequence # 3

5. Results

The visual results of the synthetic RGB videos were relatively poor quality. The output looked like a painting of a face with minor distortions rather than a photo-realistic image. We believe this is likely to be a side-effect of the architecture used for the neural network. An example of the synthetic output is shown in Figure 4.

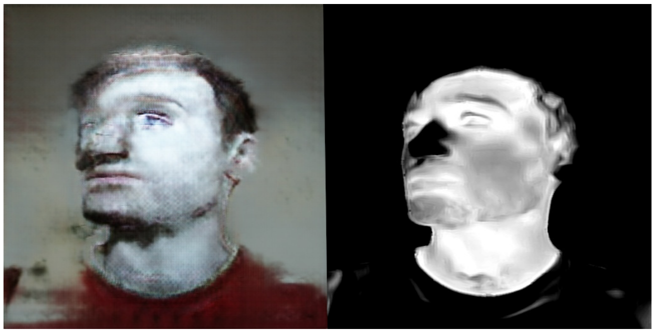

Figure 4. Comparison of the raw thermal matrix from one frame of a video and the synthetic RGB image we generated from it.

Since our application for tracking is long-term tracking, specifically for extracting vital signs from areas of human skin, we require a high robustness value as well as a high accuracy value to ensure the region of interest can be accurately tracked. It is important that both measures are sufficiently high. For instance, the robustness value may be 1.0 which means that the tracker succcessfully tracked every frame of the sequence. However, if the accuracy value is very low it suggests that the bounding box outputted by the tracker is far off from the ROI. An example of this can be seen in Figure 2 which shows the MedianFlow (blue box) and TLD (cyan box) trackers are tracking the wrong region, where the ground-truth ROI is indicated by the white box. We can also observe that these trackers often fail to recover independently and would require resetting. On the other hand, in Figure 3, we can see that the MOSSE (green box) and TLD (cyan box) trackers intersect with the ground-truth ROI which indicates high accuracy, where as the other trackers fail to accurately track the ROI.

Comparing results it is clear that CSRT was the best algorithm on every single dataset (see Table: 2). Our algorithm showed an improvement in dataset #1 and #5, and a decline in performance in #2-4 for CSRT. It is possible to make an argument that #4 was also improved if you value a high robustness of 1, since MIL was improved to an accuracy of 0.52 and robustness of 1 on the RGB, whilst CSRT on TIR had an accuracy of 0.68 which is higher, but at a robustness of 0.9.

In total using our synthetic RGB we saw an improvement on 7 out of 22 results. We can also see improvement was largely dependant on the algorithm with MIL and CSRT benefiting the most, while MedianFlow never improved.

Of important note is that dataset #4 and #5 were on dataset similar to our training data, and showed effectively best case performance, while datasets #1 through #3 were very different than our training data and could be seen as worst case performance. Therefore it would seem our algorithm did quite well, improving 1 out of 3 of the worst-case and 1 or 2 out of the 2 best cases depending on how highly you weight a high robustness metric.

This shows promise for our algorithm and we are therefore optimistic with taking this further, as with better training and design we believe we could achieve even better results. It would also be interesting to test in on a more wide variety of datasets. We should also take into account

Tracker	**Dataset #**	$Accuracy_{TIR}$	$Robustness_{TIR}$	$Accuracy_{RGB}$	$Robustness_{RGB}$	$Accuracy_{\%Change}$	$Robustness_{\%Change}$
MedianFlow	1	0.44	1.0	0.08	0.7	-82	-30
TLD	1	0.28	1.0	0.22	0.9	-21	-10
MIL	1	0.43	1.0	0.48	1.0	**+12**	**0**
MOSSE	1	0.53	0.4	0.99	0.0	+87	-100
CSRT⊙	1	0.43	1.0	0.63⊙	1.0⊙	**+47**	**0**
MedianFlow	2	0.89	0.2	0.14	0.9	-84	+350
TLD	2	0.48	1.0	0.39	1.0	-19	0
MIL	2	0.17	1.0	0.15	1.0	-12	0
MOSSE	2	0.85	0.2	0.99	0.0	+16	-100
CSRT⊙	2	0.75⊙	1.0⊙	0.16	1.0	-79	0
MedianFlow	3	0.37	0.9	0.14	0.8	-62	-11
TLD	3	0.43	1.0	0.38	1.0	-12	0
MIL	3	0.30	1.0	0.36	1.0	**+20**	**0**
MOSSE	3	0.69	0.4	0.59	0.6	**-14**	**+50**
CSRT⊙	3	0.74⊙	1.0⊙	0.65	1.0	-12	0
MedianFlow	4	0.27	0.7	0.29	0.4	+7	-43
TLD	4	0.25	1.0	0.48	1.0	**+92**	**0**
MIL	4	0.37	1.0	0.52	1.0	**+41**	**0**
CSRT⊙	4	0.68⊙	0.9⊙	0.64	0.7	-6	-22
MedianFlow	5	0.38	0.9	0.23	0.5	-39	-44
TLD	5	0.37	1.0	0.20	1.0	-46	0
CSRT⊙	5	0.46	0.4	0.55⊙	1.0⊙	**+20**	**+150**

TABLE 2. BENCHMARK RESULTS: RAW THERMAL SEQUENCE (TIR) AND SYNTHETIC RGB SEQUENCE (RGB) WITH THE ANOMALIES REMOVED. ⊙ REPRESENTS THE BEST RESULT FOR THAT DATASET. BOLD CHANGE NUMBERS INDICATE AN IMPROVEMENT USING RGB.

that our algorithm took about 0.13 seconds to process an image, therefore equating to running at about 7.3 FPS. This is almost real-time since the camera specification lists 8.7Hz as the frame rate of the camera [27]. This is slightly offset by the algorithms running quicker on the synthetic RGB than the TIR data for example TLD runs 30% faster for an average of 32.6 FPS on synthetic RGB images as opposed to 25 FPS on TIR data.

Also worth exploring are the different areas we chose for each benchmark. Dataset #1 through #3 being the worst case, had the nose chosen since it is clearly defined in the thermal data. #4 and #5 however had the mouth and nostril chosen, since these are usually hard to find in the thermal data. This makes the result on these quite key as we are getting information out of the synthetic RGB that was not there in the thermal. For example looking at figure: 4 we can see the mouth is much more defined in the synthetic RGB. We can therefore track on the RGB data for the mouth, then refer back to the raw thermal data to get the temperature if we were interested in the vital sign of breathing out of the mouth for example.

6. Discussion & Limitations

Since we found areas that were not visible in the thermal data in the RGB data, we decided to try more traditional facial tracking techniques such as landmarking instead of more general object tracking. Our result was that using a 68 point landmark tracking, specifically an implementation found in Kazemi and Sullivan [28], we had a success on 41 out of 73 frames on dataset number 5. This is in contrast to not being able to apply it at all on the thermal images with a success rate of 1 out of 73. You can see the results in figures: 5, 6. It is therefore possible we could do tracking and reinitialize the tracker based on these results and use them as initial frames, so as to limit tracking length of more general tracking to only a few frames, usually the duration where you see higher accuracy.

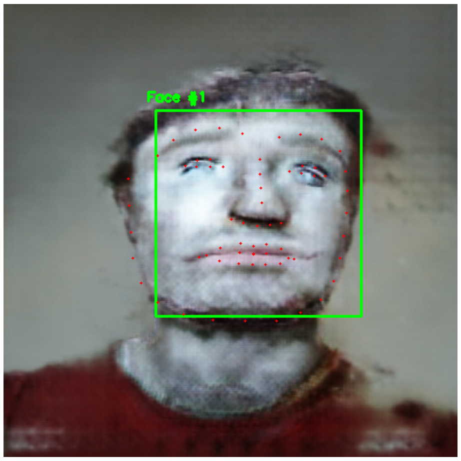

Figure 5. 68 landmark tracking for frontal profile on synthetic RGB image

A crucial feature to note however on landmark tracking success vs object tracking is that we are not accounting for occluded frames in landmark tracking. For example in figure: 6 the left eye is occluded but we have a "success" since we found the non-occluded landmarks. Therefore you would also need to process those frames to check the specific landmark you were after.

One large limitation of our approach is that our dataset was limited to images of ourselves. With a much larger and more varied data-set we would expect a large increase in quality of the synthetic images. This would also lead to a more robust and less-biased method.

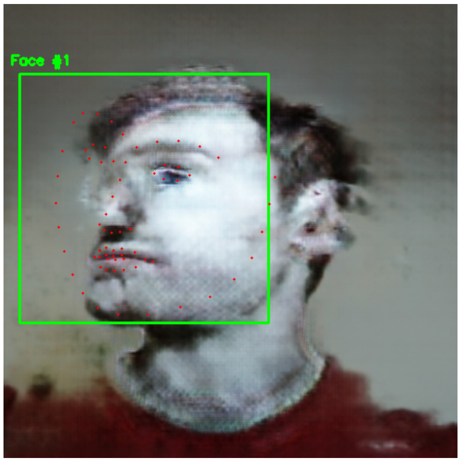

Figure 6. 68 landmark tracking for side profile on synthetic RGB image

Another limitation of our work was the loss function. Our loss function took into account the whole image, however really we don't care about re-creating the exact RGB image. All we actually want to be accurate is the facial region, so ideally our loss function could create some sort of mask, and so only take into account the accuracy of the facial region, and ignore the background and person's body since we don't care if that part is wrong. To get round this limitation in our research, we tried to make sure our training data had mostly empty backgrounds. This reduced the amount of information our neural net needed to learn. It is possible we would get better results if we used an evenly lit background and made it a singular colour.

A limitation of the landmarking approach which we also looked at was that it is trained on real RGB images, and so we need to at-least closely resemble that with our synthetic results. We therefore should see massively improved results should we get more photo-realistic images out of the network. It also doesn't take into account previous frames, as it is designed for static images and not videos.

A final more obvious limitation is that we are trying to guess or create data that isn't there in the thermal data through recognising patterns in faces. It is quite possible we need a completely different or larger architecture to approach this with a neural net to get good results. This seems less likely considering similar approaches such as pix2pixHD produce extremely photo-realistic results [29].

7. Conclusion and Future Work

In conclusion, we have demonstrated that it is possible to take a thermal matrix and to generate a synthetic RGB approximation of what it would look like. We have discovered this to be more difficult than the other way around, that is generating thermal images from RGB images. This is because there is more information to guess, since thermal images are usually quite low in high frequency detail, since heat doesn't show up as having texture or colour, whereas in RGB images there is a lot of texture, shadows and colour for the neural net to guess or "create" to get a synthetic image.

We have also demonstrated that tracking on the generated images can be better than tracking on the raw thermal video, specifically when using algorithms designed specifically for facial tracking, rather than more generalised approaches. This shows promising potential, and future work could be done on creating a better discriminator and loss function specifically for this task. That would enable a much higher quality output, which should also mean a better tracking result.

Future work could include creating a better matched thermal-RGB dataset which includes a greater variety of subjects, as-well as higher quality images, as we were stuck with a low cost flir-one camera. Potentially you could train it on high-quality images such that it would also work to convert low-quality images better. We could also use the synthetic thermal data in an RGBT tracker, instead to get the best of both worlds, these trackers are becoming more popular as shown by the VOT benchmark replacing the thermal challenge with the combined challenge.

It would also be interesting to see if you could simply generate a semantic labelling instead of an rgb and track that, say convert the eye area of the thermal image to a solid block of colour. This should be easier for the network since it doesn't have to worry about lighting, shading and other such variables.

A final promising piece of future work would be to create a tracker using a combination of generalised object tracking and specific facial tracking. Using landmark trackers where possible and a generalised object tracker for the failed frames. You would also need to analyse the "successful" frames in the landmark tracking since the specific feature you want may be occluded, but it was a success since it found other landmarks, thus marking it as absent would turn the landmark tracker into a long-term tracker.

Appendix A. Source Code of Provided Tools

A.1. FLIRONE Image Registration and Stitching Tool

This tool extracts both the thermal and rgb images from a picture taken with the FLIRONE app. It then crops and scales the rgb image to register with the thermal matrix. It can then also stitch them together to allow for easier organisation of datasets. This can be found at the following URL: https://github.com/zipy124/FLIROne-RGB-TIR-Image-Registration.

A.2. FLIRONE Thermal Matrix Video Recorder

This is an android app which can take videos with the FLIRONE camera. Unlike the app provided by FLIR it doesn't save the video in a normal video format such as .mp4 but instead as a custom file format which preserves the raw thermal matrix data. Under the FLIR SDK license agreement this cannot be published publicly until approved by FLIR. Please contact the authors for more information regarding this if needed along with code to read the file format.

A.3. Thermal Benchmarking Tool Using TIPA Library

This is a collection of functions aimed at helping you benchmark thermal trackers, using a modified thermal tracking function taken from the TIPA library. The code can be found at this URL: https://gist.github.com/zipy124/0e2301bb6791019dd9d61632be0aed8a.

Appendix B.
Dataset

Both training data and the benchmarking video of which was filmed can be made available upon request. Please contact the authors to obtain a copy.

References

[1] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, p. 1409–1422, Jul 2012. [Online]. Available: http://dx.doi.org/10.1109/tpami.2011.239

[2] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.490

[3] M. Kristan and et al., "The visual object tracking vot2017 challenge results," 10 2017, pp. 1949–1972.

[4] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1387–1395.

[5] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," 2016.

[6] Y. Hu, M. Xiao, K. Zhang, and X. Wang, "Aerial infrared target tracking in complex background based on combined tracking and detecting," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–17, 03 2019.

[7] D. A. Socolinsky and A. Selinger, "A comparative analysis of face recognition performance with visible and thermal infrared imagery," in *Object recognition supported by user interaction for service robots*, vol. 4, Aug 2002, pp. 217–222 vol.4.

[8] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, "Synthetic data generation for end-to-end thermal infrared tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 4, p. 1837–1850, Apr 2019. [Online]. Available: http://dx.doi.org/10.1109/TIP.2018.2879249

[9] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017.

[11] Y. Zhang, D. Wang, L. Wang, J. Qi, and H. Lu. (2018) Learning regression and verification networks for long-term visual tracking.

[12] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.465

[13] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, and M.-H. Yang, "Deep regression tracking with shrinkage loss," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 369–386.

[14] G. Zhu, F. Porikli, and H. Li, "Tracking randomly moving objects on edge box proposals," 2015.

[15] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukezic, A. Garcia-Martin, A. Saffari, A. Petrosino, and A. Solis Montero, "The visual object tracking vot2015 challenge results," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 564–586.

[16] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pfugfelder, L. Čehovin Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, G. Fernandez, and et al., "The sixth visual object tracking vot2018 challenge results," 2018.

[17] M. Stojanović, N. Vlahović, M. Stanković, and S. Stanković, "Object tracking in thermal imaging using kernelized correlation filters," in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, March 2018, pp. 1–6.

[18] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, Nov 2016.

[19] "Flir adas dataset," https://www.flir.co.uk/oem/adas/dataset/, accessed: 2020-03-11.

[20] "Otcvbs thermal imaging dataset," http://vcipl-okstate.org/pbvs/bench/, accessed: 2020-03-11.

[21] "Flir image extractor python library," https://github.com/nationaldronesau/FlirImageExtractor, accessed: 2020-03-11.

[22] Y. Cho, S. J. Julier, N. Marquardt, and N. Bianchi-Berthouze, "Robust tracking of respiratory rate in high-dynamic range scenes using mobile thermal imaging," *Biomed. Opt. Express*, vol. 8, no. 10, pp. 4480–4503, Oct 2017. [Online]. Available: http://www.osapublishing.org/boe/abstract.cfm?URI=boe-8-10-4480

[23] Y. Cho, "Tipa: Thermal imaging-based physiological and affective computing python library," https://github.com/deepneuroscience/TIPA, accessed: 2020-03-11.

[24] I. Guyon, "A scaling law for the validation-set training-set size ratio," in *AT & T Bell Laboratories*, 1997.

[25] M. K. et al., "The seventh visual object tracking vot2019 challenge results," 2019.

[26] "Vott (visual object tagging tool)," https://github.com/microsoft/VoTT, accessed: 2020-03-11.

[27] "Flir one pro camera listing and specifications," https://www.flir.co.uk/products/flir-one-pro/, accessed: 2020-04-13.

[28] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, 2014.

[29] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," 2017.