

Supplemental: 360-Degree Gaze Estimation in the Wild Using Multiple Zoom Scales

Ashesh

ashes276@gmail.com

Chu-Song Chen

chusong@csie.ntu.edu.tw

Hsuan-Tien Lin

htlin@csie.ntu.edu.tw

National Taiwan University

Taipei, Taiwan



Figure 1: We show 3 pre-processed image sequences for our Sequential (Video) model, one in each row. With $T=3$ set for our experiments, sequence length becomes $2T + 1 = 7$. The target frame, the frame for which we want to predict the gaze is the 4th frame. Firstly, one can notice that change in eye movement is quite significant across frames. Therefore, one would ideally want eye level details to come majorly from the target frame. This is achieved by our pre-processing technique. Note, how images are successively zoomed-in in first 4 images and then zoomed out in last 4 images in this figure. This gives maximum zoom-in effect to the 4th frame. Secondly, this specific zoom-in zoom-out ordering also implicitly encodes the information about ordering of the frames in the input sequence—the most zoomed-in frame is the target frame and the more zoomed-out an image seems with respect to the target frame, the farther apart it is from it in time. Here, Frame 4 is most zoomed-in and with respect to it, Frame 1 is more zoomed out than Frame 2 is which in turn is more zoomed out than Frame 3. This is important specifically because we want to predict gaze for the 4th frame but our aggregation technique, the spatial max-pooling does not care about ordering of the sequence.

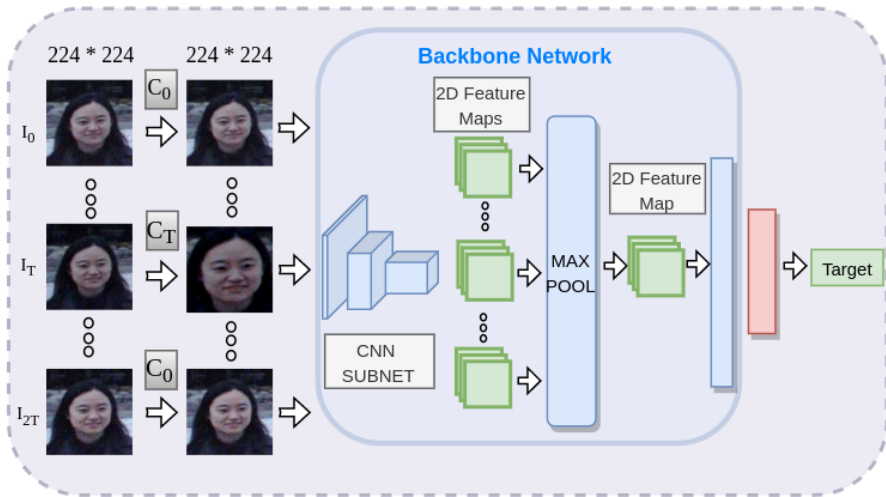


Figure 2: An end-to-end schema for Sequential (Video) model. $2T + 1$ frames are center-cropped and rescaled to original size according to the scheme mentioned in the main text. This scheme ensure greatest zoom-in affect for the middle frame. Since head pose changes are relatively minor when compared to eye movements over these frames, one would want eye orientation to get majorly captured from the middle frame, the frame for which we want to predict the gaze. Middle frame getting greatest zoom-in ensures larger and detailed availability of eye region thereby satisfying our objective.

1 Overview

Here, we briefly enumerate the content present in different sections. In Section 2, we give experiment backed intuition on why we chose the specific pre-processing scheme for our sequential (video) model MSA+Seq. In Section 3, we provide more details regarding the discontinuity in Yaw angle for full 360° yaw prediction. We also present data showing the limitation of θ_{SC} formulation due to which we came up with θ_{WSC} . In Section 4, we comment on few limitations of our work. We give implementation details in Section 5. Next, we present a couple of more experiments including ones on varying aggregation module and using LSTM in Section 6. Finally, we provide the standard error for data presented in Table 1 in the main manuscript and values and standard errors for the plots in the main manuscript in the later sections.

2 Pre-processing Scheme for Sequential Model

Our sequential (Video) model uses the MSA architecture with a specific pre-processing scheme. Let us call the frame for which we want to predict the gaze as target frame. Let us also call our proposed pre-processing scheme as ZoomIn. The scheme ensures that maximum zoom-in effect is applied on the target frame. It also ensures that the sequence order information of the frames gets implicitly encoded— prediction is to be done for most zoomed-in frame and the less zoomed-in a frame is, farther it is from the target frame in terms of sequence order. One can see few examples pre-processed with ZoomIn scheme in

Pre-processing Scheme	Angular error
None	13.0
Random	12.6
ZoomIn	12.6
Reverse	12.6

Table 1: Performance of MSA+Seq model with varying Pre-processing Scheme. First row corresponds to SSA+wavg+Seq model. We predict gaze for the *middle* frame of the sequence

Pre-processing Scheme	Angular error
Random	16.0
Reverse	13.4
ZoomIn	13.2

Table 2: Performance of MSA+Seq model with varying Pre-processing Scheme. Here, we predict the gaze for the *last* frame.

Figure 1.

The benefit of this scheme can be seen in a slightly different problem definition for video gaze prediction. In this formulation, instead of the middle frame, last frame is chosen as the target frame. In this case, according to our scheme, maximum zoom-in must be applied to the last frame. Similar to the default configuration, we have same T ($T = 3$) and CCropL. But now, given input image sequence $I_0, I_1, I_2, I_3, I_4, I_5, I_6$, we center-crop the images with sizes $C_0, C_0, C_1, C_1, C_2, C_2, C_3$ respectively and rescale them back to original size. Here, C_i is the $(i + 1)^{th}$ element in CCropL. We compare this with two other zoom-in schemes. Firstly, we use a 'Random' zoom-in scheme. In this, we permute the centercrop sizes $[C_0, C_0, C_1, C_1, C_2, C_2, C_3]$ randomly before applying them for every sequence, both in training and in evaluation. This way, the MSA model has no way to get the frame-ordering information. Secondly, we use a 'Reverse' zoom-in scheme where the target frame gets the least zoomed-in effect. In this case, given input image sequence $I_0, I_1, I_2, I_3, I_4, I_5, I_6$, we center-crop the images with sizes $C_3, C_2, C_2, C_1, C_1, C_0, C_0$ respectively and rescale them back to original size. As can be seen in Table 2, our configuration ZoomIn performs best. Significantly better performance of our ZoomIn scheme and Reverse scheme over Random scheme shows that implicitly present sequence information helps ZoomIn and Reverse schemes. Better performance of ZoomIn over Reverse scheme shows the advantage of highest zoom-in effect given to the target frame.

We also did the same experiment on our original problem definition where the target frame is the middle frame. In this case, we find that our scheme, the reverse scheme and the random scheme are all giving very similar performance as can be seen in Table 1. This has to do with how Gaze360 dataset was created. It contains mostly monotonic movement of target gaze across frames. People start from a gaze orientation and move their gaze consistently in one direction. Due to this, the average gaze of the seven frames will naturally be very close to the gaze of the middle frame, which is the target gaze. So, the network just needs to obtain average gaze of all the frames and for this, the sequence information does not matter. Relatively higher importance to the eye region of the target frame also does not matter when one aims to take the average of all gazes.

3 Discontinuity In Yaw

We want to elaborate on two points on this subject. Firstly, as stated in the main manuscript, backward gazes not only are harder to estimate owing to absence of face in the input image, but also introduce discontinuity in the yaw angle. Specifically, for small ϵ , backward gazes

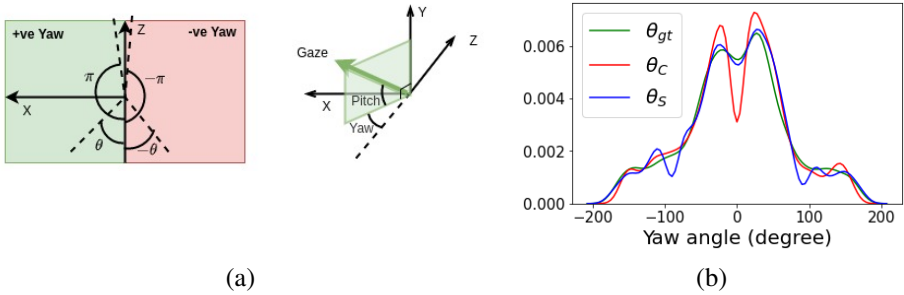


Figure 3: (a) Discontinuity in yaw space. Yaw angle defined with respect to negative z axis (right figure). Discontinuity (left figure) is seen when the projection of the gaze vector on the XZ plane is very close to the positive Z direction. From one side, the angle reaches π , and from the other, it reaches $-\pi$. (b) Distribution of actual and predicted yaw angle. Note the dip around 0° in the distribution of the cosine-based estimate θ_C

of yaw angles $\pi - \epsilon$ and $-\pi + \epsilon$ are far apart in terms of the numerical value but are of close physical proximity as can be seen in Figure 3 (a).

Secondly, in Figure 3 (b), we show the distribution of θ_C , θ_S and the ground truth θ_{gt} . One can observe a significant dip at around 0° in θ_C . This indicates that θ_C has difficulty in predicting around 0° and which arguably is the cause for inferior performance of θ_{SC} on frontal $\pm 20^\circ$. The reason for this and the remedy of using θ_{WSC} is given in the main manuscript.

4 Challenges and Limitations

We observe that the MSA model does not give benefit to backward gazes. This becomes clear when one looks at the Back column of Table 1 of the main manuscript. We feel this is intuitive. MSA aims to extract information from two magnification scales— eye region and overall head region. For backward gazes, the face is not visible— eyes, nose and other facial details are completely absent. So, one needs to estimate gaze from the overall head orientation, i.e, the head pose. As the information is not contained in multiple magnification levels, the MSA approach gives no benefit.

The other limitation of this approach is time complexity for single-image input model type. Firstly, we note that for sequential models, MSA+Seq takes same amount of time as SSA+avg+Seq and so there is no issue of time complexity here. For single-image input models, the time complexity of MSA scales linearly with the number of elements in CCropL. CCropL=[224, 200, 175, 150] takes 4 times more time than Static model. As far as the GPU RAM is concerned, MSA takes twice GPU RAM as Pinball Static [9]. For working within twice GPU RAM, one would need to compute the 2D feature map one by one while simultaneously updating the max feature map stored in a buffer. CCropL=[224, 150] takes two times more time than Static model. However, unlike LSTM, MSA can be parallelized since the 2D feature map computation for each magnification level is independent of each other and so one could easily improve upon the time complexity issue.

Centercrop size	Val error	Test error
224	13.72	13.83
210	13.68	13.78
200	13.73	13.76

Table 3: Angular error obtained on front 180° gazes in Test and Validation sets using Static+avg model with MobileNet backbone on Gaze360 dataset. At evaluation time, center-crop followed by resize operation is done.

Aggregation module	Angular error
Spatial-Max	13.9
MAX	14.1
Spatial-Attention	14.0
LSTM	14.1

Table 4: Comparison of different aggregation modules with MSA

5 Implementation Details

Similar to [9], we fixed the backbone’s output layer size to 256 for all our experiments. Our network input is 224x224x3-sized images. For having a fair comparison with [9], we also used $T = 3$ for sequence model. For all experiments on all three datasets and both model types (video and single-image), unless specified otherwise, we used [224,200,175,150] for CCropL, which we obtained empirically. When using the LSTM module as an aggregation module, following [9], we used bidirectional LSTM with two layers and a hidden size of 256. We implemented the model using PyTorch. For all of the Gaze360 experiments, we trained the network for 100 epochs with a batch size of 64, a learning rate of 0.0001, and the Adam optimizer.

For the experiments on the RT-GENE dataset [2], following their GitHub code,¹ we used a learning rate of 0.000325 and the Adam optimizer with the same hyperparameters ($\text{betas} = (0.9, 0.95)$). Additionally, we use early stopping on validation loss with a patience of 5.

6 Additional Experiments

6.1 Empirical Evidence on Significance of Scale in Gaze360 dataset

Here, we use Static+avg model with the MobileNet backbone. At evaluation time, we added center-cropping with a fixed size and subsequent resizing to the original size in the data preprocessing step for both the test and validation sets. Note that during model training, no center-cropping was performed on the training and validation sets. One therefore expects performance to degrade upon the introduction of center-cropping at evaluation time. However, we found a significant proportion of images having a better performance with the cropping based preprocessing so much so that, we observe a minute performance improvement on overall dataset as can be seen in Table 3. This implies that dataset contains images of varying scales and the model does not extract features equally efficiently from all scales. We observe similar findings on RT-GENE dataset as well whose data is presented in next subsection.

¹https://github.com/Tobias-Fischer/rt_gene/

Center-crop size	Val error (Hardnet68)			Val error (Resnet18)		
	K-fold 0	K-fold 1	K-fold 2	K-fold 0	K-fold 1	K-fold 2
224	7.21	5.68	5.86	7.67	6.14	6.38
215	7.04	5.66	5.84	7.60	6.14	6.4
210	7.00	5.68	5.88	7.62	6.20	6.49

Table 5: Angular error on Validation set using static model with Resnet18 and Hardnet backbones on RT-GENE

Centercrop size	Angular error
224	14.5 ± 0.16
200	14.31 ± 0.06
175	14.3 ± 0.13
150	14.4 ± 0.19

Table 6: Finding the optimal crop-size for Static+avg + Reg model with Resnet18 backbone.

Method	Backbone	Angular Error
MSA+raw + LSTM	Resnet	7.1
MSA+raw + LSTM	Hardnet	6.9

Table 7: Performance using LSTM as aggregation module on RT-GENE dataset using Raw-Original image type

6.2 Empirical Evidence on Significance of Scale in RT-GENE dataset

As shown in Table 5, similar to Gaze360, overall performance on the validation set improves when we add center-crop preprocessing at evaluation time for RT-GENE dataset. Being more prominent for K-fold=0, this holds true for all three folds with the Hardnet68 backbone. This indicates that (1) images in the data have multiple scales and that (2) the Static model [8] does not capture features from all scales equally well. It is worth noting that this is an ‘indicator’ experiment. For us to take cue from it, it is not necessary for this effect to manifest with all backbones to the extent that the average performance improves.

6.3 Benefit of Using Multiple-Scales Over Single Optimal Scale

Given the evidence in Subsection 6.1 showing that Gaze360 dataset has decent variations in scale, it is reasonable to assume the existence of a single centercrop size which would give better performance. We therefore also wanted to check whether our MSA+avg model performs better than Static+avg model with that optimal center crop size. To be specific, all input images will be center cropped to that crop size and they will then be subsequently rescaled to original size of 224. Rescaled images are then fed to Static+avg model. We did the experiment with Resnet18 backbone. Results present in Table 6 show that 175 as center-crop size is the optimal configuration. As can be seen from Table 1 of the main manuscript, our MSA+avg model outperforms it. Note that the comparison done this way is not fair for our MSA+avg model as one cannot know optimal centercrop size for test dataset a priori. In spite of this, this outperformance shows that using multiple cropsizes is better over the use of one crop size.

6.4 Effect of Different Aggregation Modules

Here, we varied the aggregation modules: we used Spatial-Max (used in our proposed model), LSTM, MAX and Spatial-Attention as aggregation techniques. For Spatial-Attention, 2D feature maps were aggregated using an attention module instead of max-pool. For LSTM and MAX, one-dimensional features coming out of the backbone network’s last dense layer

Name	Backbone	All 360	Front 180	Front 40	Back
Spatial Weights CNN [14]	-	0.28	0.27	0.33	0.34
Spatial Weights CNN [14]	Resnet	0.18	0.13	0.21	0.56
CA-Net [15]	-	0.54	0.49	0.34	1.77
CA-Net [15]	Resnet	0.18	0.20	0.19	0.10
Static+avg	Resnet	0.06	0.04	0.18	0.12
Static+wavg	Resnet	0.06	0.05	0.12	0.13
MSA+raw	Resnet	0.10	0.13	0.38	0.48
MSA+avg	Resnet	0.18	0.27	0.58	0.16
MSA	Resnet	0.17	0.25	0.50	0.12
MSA	Hardnet	0.08	0.13	0.31	0.089

Table 8: Standard Error of models presented in Table 1 (Left) of main manuscript

Name	Backbone	All 360	Front 180	Front 40	Back
SSA+avg	Resnet	0.05	0.04	0.06	0.36
SSA+wavg+Seq	Resnet	0.02	0.04	0.06	0.24
MSA+avg+Seq + Reg + LSTM	Resnet	0.06	0.08	0.22	0.31
MSA+Seq + Reg + LSTM	Resnet	0.05	0.07	0.17	0.29
MSA+avg+Seq	Resnet	0.06	0.05	0.12	0.19
MSA+Seq	Resnet	0.08	0.06	0.13	0.24
MSA+Seq	Hardnet	0.05	0.08	0.17	0.06

Table 9: Standard Error of models presented in Table 1 (Right) of main manuscript

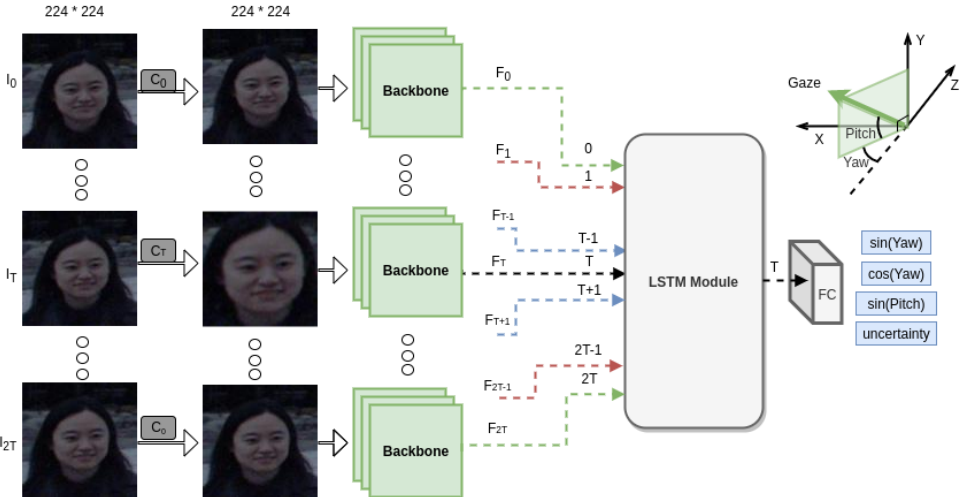


Figure 4: A variant of our Seq[W]SCMulticrop model which has LSTM as aggregation module

Aggregation Module	Standard Error
SPATIAL-MAX	0.18
MAX	0.10
SPATIAL-ATTENTION	0.08
LSTM	0.11

Table 10: Standard Error of models presented in Table 4

were aggregated. For LSTM, the input sequence ordering followed CCropL. For MAX, maximum was taken along the scale dimension. We used Resnet18 as the backbone for this study. As seen in Table 4, performance is not significantly dependant on the choice of the aggregation technique.

6.5 LSTM as an Aggregation Module

We did extensive experiments with LSTM as an aggregation module. As briefly described in the main document, the output of the backbone network is passed into the LSTM. This is different from the aggregation module for our main model MSA, since in that case, output of the CNN portion of the backbone network is aggregated. For more clarity, please refer to Figure 4. We did the experiments on both Sequence and Static model. For the Static model, output of LSTM module corresponding to last element of input sequence is taken and passed through dense layers to yield the target variables. For the Sequence model, output of the LSTM module corresponding to the middle element of the input sequence is taken and passed through the dense layers to yield the target. It is done this way for the sequence model since we want to predict the gaze for the middle frame in the input sequence.

Regularized Pinball Loss: As stated in the manuscript, following Gaze360 paper [4], we keep our loss as Pinball Loss. We explored another way here to fix model’s bad performance near $\theta = 0^\circ$. We added a constraint penalizing deviation of predicted $\sin(\theta)$, $\cos(\theta)$ from $\sin^2(\theta) + \cos^2(\theta) = 1$. It is implemented as addition of a weighted MSE loss component. Our final loss is

$$w * L_P + (1 - w) * L_{MSE}$$

, where L_P is the pinball loss, $L_{MSE} = MSE(1, \sin^2(\theta) + \cos^2(\theta))$ and w set to 0.9. We denote the presence of this regularization by “Reg” token in model name in the tables. We found this to slightly outperform with a couple of backbones (majorly with Squeezenet). One can verify the same by looking at performance data for this configuration in Tables 19, 20, 17 and 18. However, one can see that even with the regularization, the weighted sine-cosine transformation θ_{WSC} still gives better performance over naive sine-cosine transformation θ_{SC} . We, therefore did not include it in our final model configuration. Nonetheless, one can observe the benefit of using our Static[W]SCMultiCrop model over Static[W]SC model from above tables.

Performance on Static and Sequence model types are shown in Table 11 and Table 12 respectively for Gaze360 dataset. Results on RT-GENE dataset are shown in Table 7.

Model	All 360	Front 180	Front Fac-ing	Back
MSA+avg + LSTM + Reg	14.1 ± 0.10	12.3 ± 0.10	12.4 ± 0.20	20.6 ± 0.38
MSA + LSTM + Reg	14.1 ± 0.11	12.3 ± 0.10	12.3 ± 0.20	20.6 ± 0.37

Table 11: Performance comparison for Static models with LSTM as aggregation module and with Resnet backbone on Gaze360 dataset [9]

Model	All 360	Front 180	Front Fac-ing	Back
SSA+avg+Seq + LSTM + Reg	13.18 ± 0.03	11.45 ± 0.06	11.2 ± 0.11	19.4 ± 0.34
SSA+wavg+Seq + LSTM + Reg	13.12 ± 0.05	11.39 ± 0.05	10.9 ± 0.19	19.3 ± 0.35
MSA+avg+Seq + LSTM + Reg	12.74 ± 0.06	10.94 ± 0.08	10.7 ± 0.22	19.2 ± 0.32
MSA+Seq + LSTM + Reg	12.71 ± 0.05	10.91 ± 0.07	10.6 ± 0.17	19.2 ± 0.30

Table 12: Performance comparison for Sequential models with LSTM as aggregation module and with Resnet backbone on Gaze360 dataset [9]

7 Standard Error Data on Model Performances

In Table 8, 9 and 10 , we present the data containing standard error of models present in Table 1 (Left) and Table 1 (Right) of of the main manuscript and Table 4 respectively. We train the models 3 times and compute the standard deviation of mean angular error.

8 Performance Data on Different Backbones

In Tables 13,14, 15 and 16 we present the average angular error along with standard error of our different static models over multiple backbones. Note that for generating this data, each model configuration was trained three times independently and mean and the standard deviation of the three average angular error numbers are reported.

Backbone	Static	Static+avg	Static+wavg	MSA+avg	MSA
SqueezeNet	20.9 ± 0.89	17.8 ± 0.10	17.49 ± 0.03	20.2 ± 0.18	19.62 ± 0.37
Shufflenet	17.96 ± 0.06	17.25 ± 0.04	17.12 ± 0.07	16.1 ± 0.11	15.9 ± 0.12
Mobilenet	16.3 ± 0.19	15.8 ± 0.47	15.7 ± 0.41	14.6 ± 0.13	14.5 ± 0.10
Resnet18	15.78 ± 0.07	14.44 ± 0.06	14.35 ± 0.06	14.0 ± 0.18	13.9 ± 0.17
Hardnet68	14.7 ± 0.18	13.91 ± 0.05	13.85 ± 0.06	13.8 ± 0.10	13.72 ± 0.08

Table 13: Model Performances (with standard error) on all 360° gazes

Backbone	Static	Static+avg	Static+wavg	MSA+avg	MSA
Squeezenet	15.2 ± 0.46	15.4 ± 0.17	15.12 ± 0.08	17.8 ± 0.28	17.3 ± 0.30
Shufflenet	15.2 ± 0.10	15.55 ± 0.04	15.42 ± 0.07	14.13 ± 0.06	14.02 ± 0.07
Mobilenet	13.8 ± 0.19	14.1 ± 0.53	14.0 ± 0.49	12.9 ± 0.10	12.8 ± 0.12
Resnet18	13.0 ± 0.13	12.77 ± 0.04	12.69 ± 0.05	12.3 ± 0.27	12.20 ± 0.25
Hardnet68	12.4 ± 0.10	12.3 ± 0.10	12.2 ± 0.11	12.10 ± 0.14	12.03 ± 0.13

Table 14: Model Performances (with standard error) on Front 180° gazes

Backbone	Static	Static+avg	Static+wavg	MSA+avg	MSA
Squeezenet	13.3 ± 0.56	16.5 ± 0.96	13.6 ± 0.1	20.6 ± 0.62	13.5 ± 0.25
Shufflenet	15.1 ± 0.1	16.45 ± 0.09	15.63 ± 0.06	14.2 ± 0.18	13.2 ± 0.16
Mobilenet	13.79 ± 0.05	14.48 ± 0.44	13.68 ± 0.12	12.9 ± 0.22	12.27 ± 0.19
Resnet18	13.1 ± 0.25	13.1 ± 0.18	12.8 ± 0.12	12.6 ± 0.58	12.1 ± 0.50
Hardnet68	12.3 ± 0.37	12.4 ± 0.11	12.08 ± 0.11	12.1 ± 0.40	11.67 ± 0.30

Table 15: Model Performances (with standard error) on front facing (40°) gazes

Backbone	Static	Static+avg	Static+wavg	MSA+avg	MSA
Squeezenet	41 ± 2.7	26.3 ± 0.24	26.0 ± 0.34	28.9 ± 0.61	28.2 ± 0.54
Shufflenet	28.0 ± 0.43	23.1 ± 0.5	23.2 ± 0.13	23.0 ± 0.50	22.9 ± 0.58
Mobilenet	25.5 ± 0.35	22.0 ± 0.76	21.7 ± 0.7	20.9 ± 0.29	20.6 ± 0.25
Resnet18	25.6 ± 0.21	20.4 ± 0.12	20.4 ± 0.13	19.9 ± 0.16	19.9 ± 0.12
Hardnet68	22.9 ± 0.50	19.7 ± 0.21	19.6 ± 0.17	19.98 ± 0.04	19.81 ± 0.09

Table 16: Model Performances (with standard error) on back gazes

Model	All 360	Front 180	Front Facing	Back
Squeezenet	15.94	13.05	12.85	26.32
Shufflenet	16.11	14.31	14.94	22.61
Mobilenet	14.71	12.86	12.87	21.34
Resnet18	14.07	12.19	12.14	20.86
Hardnet68	13.65	11.98	11.93	19.69

Table 17: Angular error with LSTM as aggregation module for MSA+avg+LSTM+REG model

Model	All 360	Front 180	Front Facing	Back
Squeezenet	15.91	13.03	12.64	26.27
Shufflenet	16.07	14.26	14.67	22.58
Mobilenet	14.66	12.83	12.71	21.24
Resnet18	14.05	12.16	12.04	20.84
Hardnet68	13.63	11.95	11.85	19.66

Table 18: Angular error with LSTM as aggregation module for MSA+LSTM+REG model

Model	All 360	Front 180	Front Facing	Back
Squeezenet	16.72	14.26	14.74	25.57
Shufflenet	17.52	15.91	16.49	23.3
Mobilenet	15.48	13.74	13.14	21.76
Resnet18	14.41	12.66	12.81	20.72
Hardnet68	13.81	12.19	12.29	19.64

Table 19: Angular error with LSTM as aggregation module for Static+avg+REG model

Model	All 360	Front 180	Front Facing	Back
Squeezenet	16.68	14.26	12.58	25.37
Shufflenet	17.43	15.83	15.81	23.19
Mobilenet	15.43	13.7	12.95	21.62
Resnet18	14.35	12.6	12.5	20.65
Hardnet68	13.75	12.14	12.07	19.54

Table 20: Angular error with LSTM as aggregation module for Static+wavg+REG model

References

- [1] Yihua Cheng, Shiyao Huang, Fei Wang, Chen Qian, and Feng Lu. A Coarse-to-Fine Adaptive Network for Appearance-Based Gaze Estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):10623–10630, April 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i07.6636. URL <https://aaai.org/ojs/index.php/AAAI/article/view/6636>.
- [2] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 339–357, Cham, 2018. Springer International Publishing.
- [3] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically Unconstrained Gaze Estimation in the Wild. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6911–6920, October 2019.
- [4] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. It’s Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2299–2308, July 2017.