

SUPPLEMENTAL MATERIAL FOR LABEL ENCODING FOR REGRESSION NETWORKS

Deval Shah, Zi Yu Xue & Tor M. Aamodt

Department of Electrical and Computer Engineering
University of British Columbia, Vancouver, BC, Canada
{devalshah, fzyxue, aamodt}@ece.ubc.ca

The appendices in this supplemental material provide an ablation study (Appendix A), derivation of equations used for expected error analysis (Appendix B-C), a detailed description of benchmarks and evaluation (Appendix D). Code is available at https://github.com/ubc-aamodt-group/BEL_regression.

A ABLATION STUDY

Impact of combination of encoding, decoding, and loss functions: We propose multiple combinations of encoding, decoding, and loss functions that can be used with BEL. In Tables 1- 11, we show the effect of each combination of encoding, decoding, and loss function on the error of the model. Although general trends exist and some combinations perform consistently well across datasets, the optimal combination varies based on the dataset.

Table 1: Comparison of BEL design parameters on MAE for head pose estimation with BIWI dataset and ResNet50 feature extractor (HPE1).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	3.38	3.65	-	-	-	-
GEN-EX	BCE	3.37	3.64	5.11	8.02	4.76	7.53
GEN	BCE	3.38	3.65	5.16	8.16	4.99	7.73
GEN-EX	CE	4.22	3.55	3.88	4.08	4.09	5.50
GEN	CE	4.25	3.62	3.93	4.06	4.39	5.48
GEN-EX	L2	3.56	3.93	3.66	3.59	5.99	4.21

Table 2: Comparison of BEL design parameters on MAE for head pose estimation with 300LP/AFLW2000 datasets and ResNet50 feature extractor (HPE2).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	4.78	4.84	-	-	-	-
GEN-EX	BCE	4.77	4.84	5.43	5.09	4.94	7.84
GEN	BCE	4.78	4.87	5.11	5.05	5.15	8.54
GEN-EX	CE	4.93	5.04	5.04	4.97	4.79	5.64
GEN	CE	5.07	5.17	5.13	5.10	4.99	5.62
GEN-EX	L2	5.05	5.18	5.19	5.09	5.17	5.07

Table 3: Comparison of BEL design parameters on MAE for head pose estimation with BIWI dataset and RAFA-Net feature extractor (HPE3).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	3.47	3.16	-	-	-	-
GEN-EX	BCE	3.46	3.12	3.30	3.35	3.80	5.75
GEN	BCE	3.49	3.14	3.62	3.78	4.44	5.83
GEN-EX	CE	3.82	3.91	3.52	3.49	3.98	3.98
GEN	CE	3.92	4.09	3.62	3.65	4.35	4.28
GEN-EX	L2	3.72	3.60	4.31	4.29	6.61	18.69

Table 4: Comparison of BEL design parameters on MAE for head pose estimation with 300LP/AFLW2000 datasets and RAFA-Net feature extractor (HPE4).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	3.94	4.00	-	-	-	-
GEN-EX	BCE	3.90	3.93	4.19	4.12	4.39	9.17
GEN	BCE	3.93	3.94	4.21	4.25	4.53	9.21
GEN-EX	CE	4.55	4.62	4.34	4.53	4.45	5.12
GEN	CE	4.68	4.75	4.46	4.61	4.63	5.29
GEN-EX	L2	4.45	5.87	5.11	9.34	10.43	17.89

Table 5: Comparison of BEL design parameters on NME for facial landmark detection with COFW dataset and HRNetV2-W18 feature extractor (FLD1).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	3.47	3.45	-	-	-	-
GEN-EX	BCE	3.45	3.43	3.42	3.41	3.47	4.28
GEN	BCE	3.46	3.45	3.43	3.47	3.66	4.43
GEN-EX	CE	3.37	3.37	3.38	3.41	3.34	3.69
GEN	CE	3.44	3.44	3.44	3.49	3.57	3.69
GEN-EX	L1	3.44	3.41	3.45	3.47	3.41	4.52

Table 6: Comparison of BEL design parameters on NME for facial landmark detection with 300W dataset and HRNetV2-W18 feature extractor (FLD2).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	3.5	3.49	-	-	-	-
GEN-EX	BCE	3.48	3.46	3.43	3.42	3.38	4.71
GEN	BCE	3.50	3.49	3.45	3.45	3.55	4.78
GEN-EX	CE	3.40	3.36	3.37	3.41	3.37	3.62
GEN	CE	3.50	3.45	3.45	3.51	3.59	3.65
GEN-EX	L1	3.41	3.39	3.49	3.67	3.43	4.04

Table 7: Comparison of BEL design parameters on NME for facial landmark detection with WFLW dataset and HRNetV2-W18 feature extractor (FLD3).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	4.62	4.54	-	-	-	-
GEN-EX	BCE	4.6	4.51	4.43	4.38	4.37	7.18
GEN	BCE	4.62	4.53	4.44	4.42	4.55	7.14
GEN-EX	CE	4.36	4.34	4.36	4.33	4.34	5.15
GEN	CE	4.46	4.44	4.47	4.47	4.56	4.83
GEN-EX	L1	4.39	4.42	4.47	4.47	4.45	4.74

Table 8: Comparison of BEL design parameters on NME for facial landmark detection with AFLW dataset and HRNetV2-W18 feature extractor (FLD4).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	1.51	1.52	-	-	-	-
GEN-EX	BCE	1.50	1.50	1.47	1.47	1.49	1.52
GEN	BCE	1.51	1.52	1.50	1.49	1.54	1.55
GEN-EX	CE	1.48	1.47	1.47	1.47	1.47	1.47
GEN	CE	1.52	1.51	1.51	1.51	1.52	1.52
GEN-EX	L1	1.47	1.47	1.48	1.48	1.48	1.59

Table 9: Comparison of BEL design parameters on MAE for age estimation with MORPH-II dataset and ResNet50 feature extractor (AE1).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	2.32	2.27	-	-	-	-
GEN-EX	BCE	2.30	2.29	2.35	2.49	2.45	2.99
GEN	BCE	2.28	2.28	2.34	2.51	2.54	3.07
GEN-EX	CE	2.55	2.54	2.75	2.65	2.63	12.33
GEN	CE	2.60	2.58	2.61	2.66	2.61	3.10
GEN-EX	L1	2.30	2.30	2.32	2.30	2.32	2.29

Table 10: Comparison of BEL design parameters on MAE for age estimation with AFAD dataset and ResNet50 feature extractor (AE2).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	3.13	3.15	-	-	-	-
GEN-EX	BCE	3.14	3.16	3.32	3.35	3.28	3.34
GEN	BCE	3.13	3.19	3.41	3.44	3.41	3.52
GEN-EX	CE	3.26	3.29	3.38	3.44	3.40	3.30
GEN	CE	3.36	3.34	3.42	3.47	3.40	3.45
GEN-EX	L1	3.13	3.12	3.11	3.12	3.13	3.13

Table 11: Comparison of BEL design parameters on MAE for end-to-end autonomous driving with PilotNet dataset and feature extractor (PN).

Decoding function	Loss function	Encoding function					
		U	J	B1JDJ	B2JDJ	HEXJ	HAD
BEL-J/BEL-U	BCE	4.34	3.91	-	-	-	-
GEN-EX	BCE	4.57	4.20	4.83	4.96	5.29	10.12
GEN	BCE	4.37	3.95	3.51	3.61	4.01	10.00
GEN-EX	CE	4.30	4.16	4.99	5.87	5.39	87.17
GEN	CE	3.15	3.11	3.14	3.21	3.64	6.20
GEN-EX	L1	4.10	4.11	4.34	4.34	4.11	5.09

Impact of quantization and decoding functions: As discussed in Section 3, a real-valued label is quantized to a discrete value in $\{1, 2, \dots, N\}$ before applying the encoding function. In Table 12, we show the effect of increasing the number of quantization levels (N) on the error for correlation-based decoding (\mathcal{D}^{GEN} , which returns a quantized prediction) and expected correlation-based decoding ($\mathcal{D}^{\text{GEN-EX}}$, which returns a continuous prediction). As shown in the table, there exists a tradeoff between reducing quantization error and using fewer classifiers. The error is lower for 128 quantization levels than it is for 256 as the improvement resulting from fewer binary classifiers is higher than the increase in quantization error. Moreover, the use of proposed decoding function $\mathcal{D}^{\text{GEN-EX}}$ for regression consistently results in lower error compared to \mathcal{D}^{GEN} .

Table 12: Impact of the quantization and decoding functions on NME for facial landmark detection.

Quantization levels	COFW			300W		
	64	128	256	64	128	256
$\mathcal{E}^{\text{BEL-U}} + \mathcal{D}^{\text{GEN}}$	3.66	3.51	3.46	3.79	3.59	3.46
$\mathcal{E}^{\text{BEL-U}} + \mathcal{D}^{\text{GEN-EX}}$	3.46	3.41	3.44	3.54	3.47	3.44
$\mathcal{E}^{\text{BEL-J}} + \mathcal{D}^{\text{GEN}}$	3.65	3.49	3.43	3.76	3.58	3.46
$\mathcal{E}^{\text{BEL-J}} + \mathcal{D}^{\text{GEN-EX}}$	3.45	3.40	3.42	3.52	3.45	3.43

Impact of the number of training samples on BEL: As discussed in Section 4, the performance of different encoding functions varies depending on the availability of sufficient training data. In Table 13, we analyze the effect of the number of available training samples for both simple and complex encodings. We use the number of bit transitions as a measure of the complexity of a classifier. As the number of training samples decreases, simpler encodings (U and J) perform better than more complex encodings (B1JDJ, B2JDJ, and HEXJ). Using a more complex encoding reduces the number of classifiers; however, it increases each classifier’s complexity (i.e. the number of bit transitions) and thus performs poorly with less training data.

Table 13: Effect of training dataset size on optimal encoding function for facial landmark detection. BCE loss function and GEN-EX decoding function are used for the training and evaluation.

Encoding	#Classifiers/label	#bit transitions/classifier	Reduction in the number training samples						
			0%	20%	40%	60%	80%	90%	95%
COFW (FLD1)									
U	256	1	3.45	3.48	3.55	3.72	3.94	4.52	6.29
J	128	2	3.43	3.48	3.51	3.61	3.88	4.32	5.39
B1JDJ	65	4	3.42	3.44	3.52	3.60	4.11	4.50	5.68
B2JDJ	34	8	3.41	3.45	3.48	3.80	3.94	4.80	6.56
HEXJ	17	32	3.47	3.69	3.78	4.03	4.61	5.48	6.69
300W (FLD2)									
U	256	1	3.48	3.55	3.58	3.64	3.89	4.26	5.66
J	128	2	3.46	3.56	3.52	3.58	3.79	4.04	4.58
B1JDJ	65	4	3.43	3.48	3.53	3.61	3.89	4.31	6.10
B2JDJ	34	8	3.42	3.47	3.51	3.54	3.88	4.50	5.80
HEXJ	17	32	3.38	3.64	3.73	3.97	4.41	5.38	6.60
WFLW (FLD3)									
U	256	1	4.60	4.67	4.83	5.00	5.37	6.04	7.46
J	128	2	4.51	4.60	4.65	4.84	5.23	5.64	6.39
B1JDJ	65	4	4.43	4.44	4.52	4.66	5.08	5.90	8.39
B2JDJ	34	8	4.38	4.46	4.49	4.61	5.02	5.95	8.78
HEXJ	17	32	4.37	4.60	4.72	4.96	5.72	6.86	8.09
AFLW (FLD4)									
U	256	1	1.50	1.53	1.53	1.56	1.61	1.68	1.83
J	128	2	1.50	1.51	1.52	1.54	1.60	1.68	1.79
B1JDJ	65	4	1.47	1.50	1.52	1.54	1.60	1.67	1.78
B2JDJ	34	8	1.47	1.50	1.50	1.52	1.57	1.64	1.73
HEXJ	17	32	1.49	1.54	1.54	1.55	1.59	1.71	1.89

Impact of reflected binary conversion: As mentioned in Section 3, we use reflected binary to increase the distance between distant labels based on the design properties of suitable regression encodings we proposed. Table 14 shows the impact of using reflected binary conversion on error for

facial landmark detection benchmarks. As shown in the table, the use of reflected binary significantly reduces the error.

Table 14: Effect of reflected binary conversion for B1JDJ encoding on facial landmark detection. Here, BCE loss and GEN-EX decoding functions are used.

	COFW	300W	WFLW	AFLW
B1JDJ	3.43	3.46	4.43	1.47
B1JDJ- w/o reflected binary	4.13	4.43	5.70	1.97

Use of binary heamaps: Facial landmark detection approaches typically use heatmap regression. We also evaluate BEL-H-x, in which the real-valued heatmaps are converted to binary heatmaps with 8 quantization levels. Table 15 shows the impact of using binary heatmaps on error for facial landmark detection benchmarks. For unary code, a 64×64 real-valued heatmap of one facial landmark is converted to eight 64×64 binary heatmaps, resulting in 32,768 ($8 \times 64 \times 64$) binary classifiers compared to 512 for BEL-U. We believe that training a high number of binary classifiers results in high error for BEL-H-x.

Table 15: Comparison of BEL with heatmaps for facial landmark detection. Here, BCE loss and GEN-EX decoding functions are used.

	FLD1	FLD2	FLD3	FLD4
BEL-U	3.45	3.46	4.60	1.50
BEL-J	3.48	3.46	4.51	1.50
BEL-H-U	4.13	4.43	5.70	1.97
BEL-H-J	10.17	33.02	22.50	2.99

Hyperparameter θ : As shown in Figure 4b, we introduce a feature vector of size θ before the output layer. Figure 1 compares the decrease in the error for different encodings and θ values. We observe that more complex encodings benefit more from an increase in the value of θ , while a lower value of θ can be used for simpler encodings.

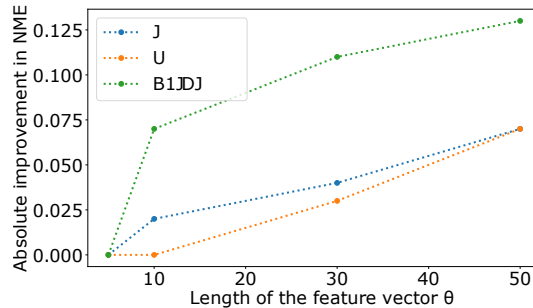


Figure 1: Effect of θ on error for different encodings on FLD1.

Impact of increasing the number of fully connected layers: For BEL, we propose to add a fully connected bottleneck layer in the regressor to reduce the feature vector size to θ and thus decrease the number of parameters in the regressor. We perform an ablation study to study the impact of this added fully connected layer on relative performance of direct regression, multiclass classification, and binary encoded labels. Table 16 provides the error (MAE or NME) for direct regression and multiclass classification with one or two fully connected layers after the feature extractor. Further, we evaluate BEL, direct regression, and multiclass classification for higher number of fully connected layers as shown in Table 17. We observe that increasing the number of fully connected layers in direct regression and multiclass classification does not improve the accuracy for most benchmarks (possibly due to overparameterization). BEL with two fully connected layers outperforms direct regression and multiclass classification in both cases. Furthermore, even for a higher number of fully connected

Table 16: Impact increasing number of fully connected layers in direct regression and multiclass classification on the error (MAE or NME).

Benchmark	Direct regression		Multiclass classification		BEL
	1 FC layer	2 FC layers	1 FC layer	2 FC layers	2 FC layers
HPE1	4.76	5.19	4.49	4.82	3.37
HPE2	5.65	5.59	5.31	5.42	4.77
HPE3	3.40	3.54	4.45	4.54	3.12
HPE4	4.14	4.22	5.14	5.45	3.90
FLD1	3.60	3.63	3.58	3.56	3.34
FLD2	3.54	3.58	3.51	3.62	3.36
FLD3	4.64	4.63	4.50	4.64	4.33
FLD4	1.51	1.51	1.56	1.53	1.47
AE1	2.44	2.35	2.75	2.81	2.27
AE2	3.21	3.14	3.38	3.40	3.11
PN	4.24	4.33	4.56	5.74	3.11

layers in BEL, the suitability of an encoding function varies with the dataset, demonstrating the importance of BEL design space.

Table 17: Impact increasing number of fully connected layers in direct regression, multiclass classification, and BEL. GEN-EX decoding function and BCE loss function are used for BEL.

Benchmark	# FC layers (size of FC layers)	Direct regression	Multiclass classification	U	J	B1JDJ	B2JDJ	HEXJ
FLD1	1 (1024-x)	3.6	3.58	-	-	-	-	-
	2 (1024-30-x)	3.63	3.56	3.45	3.43	3.42	3.41	3.47
	3 (1024-30-10-x)	3.63	3.94	3.55	3.47	3.82	4.02	3.62
FLD2	1 (1024-x)	3.54	3.51	-	-	-	-	-
	2 (1024-10-x)	3.58	3.62	3.48	3.46	3.43	3.42	3.38
	3 (1024-30-10-x)	3.55	3.78	3.42	3.46	3.5	3.61	3.52

Training-validation set based evaluation: Ideally, a validation set should be used for model selection. Hence we have reevaluated the benchmarks with a validation set to select the best design parameters and the best model (i.e., which model is the best over multiple epochs). Since datasets used in benchmarks do not provide separate validation datasets, we use 20% of the training data as a validation set. Since earlier works use 100% training data for the reported results and use test error for model selection, we have re-run specialized approaches (if possible), direct regression, and multiclass classification. It was not possible for us to re-run experiments for all specialized approaches due to resource constraints, and the comparison is conservative for many benchmarks.

Table 18 compares different regression approaches for this evaluation setup. Note that the additional results do not diminish the effectiveness of BEL and BEL outperforms direct regression and multiclass classification for all benchmarks and specialized approaches for several benchmarks.

B EXPECTED ERROR DERIVATION

This section explains the expected error equations used to compare BEL-U and BEL-J in Section 3. We first explain the encoding and decoding function used for BEL-U and derive the relation between the expected regression error and classification error for BEL-U. Then, we explain the encoding/decoding functions and expected error relation for BEL-J.

B.1 PRELIMINARIES

Given a sample i drawn from a dataset with minimum label a and maximum label b , let $y_i \in [a, b]$ represent the target label for that sample. Assuming uniform quantization, the range of target labels can be quantized using $q : [a, b] \rightarrow \{1, 2, \dots, N\}$ through Equation 1.

$$q(y_i) = (y_i - a) * \frac{N - 1}{b - a} + 1 \quad (1)$$

Table 18: Comparison of BEL with different regression approaches. Specialized approaches for each benchmark are described in Table 1

Approach (% training set)	Error (MAE or NME) / Model size			
	HPE1	HPE2	HPE3	HPE4
Specialized approach (100%)	-	-	3.40 / 69.8M	4.14 / 69.8M
Specialized approach (80%)	-	-	4.08±0.11 / 69.8M	4.69±0.02 / 69.8M
Direct regression (80%)	6.12±0.02 / 23.5M	5.97±0.09 / 23.5M	4.08±0.11 / 69.8M	4.67±4.70 / 69.8M
Multiclass classification (80%)	5.38±0.03 / 24.2M	5.60±0.13 / 24.8M	5.58±0.04 / 72.0M	5.86±0.10 / 72.0M
BEL (80%)	3.91 ±0.08 / 23.6M	4.91 ±0.10 / 23.6M	3.50±0.08 / 69.8M	3.99 ±0.04 / 69.8M
BEL $\mathcal{E}/\mathcal{D}/\mathcal{L}$ functions	U/GEN-EX/L2	U/GEN-EX/BCE	B1JDJ/GEN-EX/BCE	U/GEN-EX/BCE

Approach (%training set)	Error (MAE or NME) / Model size			
	FLD1	FLD2	FLD3	FLD4
Specialized approach (100%)	3.45 / 9.6M	3.32 / 9.6M	4.32 / 9.6M	1.57 / 9.6M
Direct regression (80%)	3.70±0.04 / 10.2M	3.69±0.06 / 10.2M	4.71±0.02 / 10.2M	1.51±0.01 / 10.2M
Multiclass classification (80%)	3.64±0.02 / 25.4M	3.68±0.02 / 45.2M	4.77±0.02 / 61.3M	1.56 ±0.01 / 20.1M
BEL (80%)	3.35 ±0.02 / 10.6M	3.40±0.03 / 11.2M	4.37±0.01 / 11.7M	1.48 ±0.01 / 10.8M
BEL $\mathcal{E}/\mathcal{D}/\mathcal{L}$ functions	HEXJ/GEN-EX/CE	U/GEN-EX/CE	B1JDJ/GEN-EX/CE	B1JDJ/GEN-EX/CE

Approach (% training set)	Error (MAE or NME) / Model size		
	AE1	AE2	PN
Specialized approach (100%)	2.49 / 21.3M	3.47 / 21.3M	4.24 / 1.8M
Direct regression (80%)	2.45 ±0.01 / 23.1M	3.34 ±0.02 / 23.1M	4.56 ±0.45 / 1.8M
Multiclass classification (80%)	2.85 ±0.03 / 23.1M	3.47 ±0.05 / 23.1M	6.37 ±0.00 / 1.9M
BEL (80%)	2.36 ±0.01 / 23.1M	3.20 ±0.00 / 23.1M	3.49 ±0.01 / 1.8M
BEL $\mathcal{E}/\mathcal{D}/\mathcal{L}$ functions	J/BEL-J/BCE	B1JDJ/GEN-EX/L1	J/GEN/CE

We define the encoding function $\mathcal{E} : \{1, 2, \dots, N-1\} \rightarrow \{0, 1\}^M$ to convert a target quantized level $Q_i \in \{1, 2, \dots, N-1\}$ to a binary code $B_i \in \{0, 1\}^M$. We further define the decoding function $\mathcal{D} : \{0, 1\}^M \rightarrow [a, b]$ to convert the predicted binary code \hat{B}_i to the predicted label \hat{y}_i .

Although the decoding functions used in this analysis predict the quantized label and introduce quantization error, we do not include quantization error in the expected absolute error for our analysis as it is constant for both BEL-U and BEL-J. The expected value of absolute error between the target y and predicted labels \hat{y} is used for the analysis as typically mean absolute error is used as the evaluation metric in regression problems.

Let us denote the error probability of a binary classifier C^k used to predict bit k in a binary code $B_i = \mathcal{E}(n)$ as $e_k(n)$, where n is the target quantized label Q_i . Then,

$$\begin{aligned}
 e_k(n) &= \mathbb{E}(|\hat{b}_i^k - b_i^k|) \\
 &= Pr(\hat{b}_i^k = F) \text{ for target label } Q_i = n \text{ and target binary code } B_i = \mathcal{E}(n)
 \end{aligned} \tag{2}$$

where $\hat{b}_i^k = T$ indicates a correct binary classification by classifier C^k ($\hat{b}_i^k = b_i^k$) for sample i and $\hat{b}_i^k = F$ indicates an incorrect binary classification by classifier C^k ($\hat{b}_i^k \neq b_i^k$) for sample i .

B.2 EXPECTED ERROR FOR BEL-U

Encoding and decoding functions: The encoding and decoding functions for BEL-U are defined as:

$$\mathcal{E}^{\text{BEL-U}}(Q_i) = b_i^1, b_i^2, \dots, b_i^{N-2}, \text{ where } b_i^k = \begin{cases} 1, & k < Q_i \\ 0, & \text{Otherwise} \end{cases} \tag{3}$$

$$\mathcal{D}^{\text{BEL-U}}(\hat{b}_i^1, \hat{b}_i^2, \dots, \hat{b}_i^{N-2}) = \sum_{k=1}^{N-2} \hat{b}_i^k + 1 \tag{4}$$

Expected error: For target quantized label $Q_i = n$ ($n \in \{1, 2, \dots, N-1\}$), ignoring the quantization error, the expected error between target y_i and predicted label \hat{y}_i can be derived as:

$$\begin{aligned}
\mathbb{E}(|\hat{y}_i^{\text{BEL-U}} - y_i|) &= \mathbb{E}\left(\left|\sum_{k=1}^{N-2} (\hat{b}_i^k + 1) - \sum_{k=1}^{N-2} (b_i^k + 1)\right|\right) \\
&= \mathbb{E}\left(\left|\sum_{k=1}^{N-2} (\hat{b}_i^k - b_i^k)\right|\right) \\
&\leq \mathbb{E}\left(\sum_{k=1}^{N-2} |\hat{b}_i^k - b_i^k|\right) \\
&= \sum_{k=1}^{N-2} \mathbb{E}|\hat{b}_i^k - b_i^k| \\
&= \sum_{k=1}^{N-2} e_k(n) \text{ (using Equation 2)}
\end{aligned} \tag{5}$$

For a uniform distribution of target labels in the range $[1, N-1]$, the expected error can be derived as:

$$\mathbb{E}(|\hat{y}^{\text{BEL-U}} - y|) \leq \frac{1}{N-1} \sum_{n=1}^{N-1} \sum_{k=1}^{N-2} e_k(n) \tag{6}$$

B.3 EXPECTED ERROR FOR BEL-J

Encoding and decoding functions: For target quantized label $Q_i \in \{1, 2, \dots, N-1\}$, BEL-J encoding requires $\frac{N}{2}$ bits/binary classifiers. The encoding for BEL-J can be defined as:

$$\mathcal{E}^{\text{BEL-J}}(Q_i) = b_i^1, b_i^2, \dots, b_i^{\frac{N}{2}}, \text{ where } b_i^k = \begin{cases} 1, & \frac{N}{2} - Q_i < k \leq N - Q_i \\ 0, & \text{Otherwise} \end{cases} \tag{7}$$

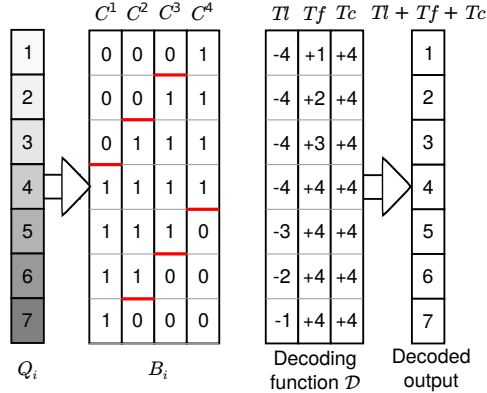


Figure 2: Encoding and Decoding functions' output for BEL-J approach and label $y \in [1, N-1]$, where $N = 8$. Decoding function's output is calculated using $y' = Tl + Tf + Tc$, where $Tl = -\max_{k \in \{1 \dots \frac{N}{2}\}} k \hat{b}_i^k$, $Tf = \max_{k \in \{1 \dots \frac{N}{2}\}} \left(\frac{N}{2} - k + 1\right) \hat{b}_i^k$, and $Tc = \frac{N}{2}$.

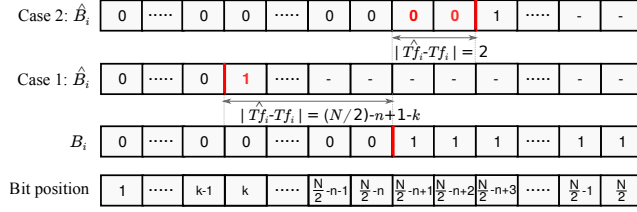


Figure 3: Effect of classifier error on $\hat{T}f_i - Tf_i$ for label $Q_i = n$. Case 1 and case 2 represent erroneous outputs. 0/1 highlighted in red color represents an error in the classifier’s output. “-” represents error/no error in both cases.

Similarly, the decoding functions for BEL-J can be defined as:

$$\begin{aligned} \mathcal{D}^{\text{BEL-J}}(\hat{B}_i) &= Tl(\hat{B}_i) + Tf(\hat{B}_i) + Tc \\ \text{where, } Tl(\hat{B}_i) &= - \max_{k \in \{1 \dots \frac{N}{2}\}} k \hat{b}_i^k \\ Tf(\hat{B}_i) &= \max_{k \in \{1 \dots \frac{N}{2}\}} \left(\frac{N}{2} - k + 1 \right) \hat{b}_i^k, Tc = \frac{N}{2} \end{aligned} \quad (8)$$

In Equation 8, $Tl()$ finds the location of the last occurrence of “1” in the predicted binary code \hat{B}_i . Similarly, $Tf()$ finds the location of the first occurrence of “1” in the binary code \hat{B}_i . Figure 2 gives examples of binary codes for label $Q_i \in \{1, 2, \dots, 7\}$ and the corresponding values of the different terms in Equation 8. For example, for label $Q_i = 3$, the binary code is “0111”. Here, the last occurrence of “1” is at position 4, and $Tl = -4$. Similarly, the first occurrence of “1” is at position 2, and $Tf = (4 + 1) - 2 = 3$.

Expected error: For BEL-J code, binary classifiers ($C^1, C^2, \dots, C^{\frac{N}{2}}$) are used. For a given input sample i , an error in any of the binary classifiers’ outputs ($\hat{b}_i^1, \hat{b}_i^2, \dots, \hat{b}_i^{\frac{N}{2}}$) will result in an error between $Tf(\hat{B}_i)/Tl(\hat{B}_i)$ and $Tf(B_i)/Tl(B_i)$ in Equation 8. We refer to $Tf(\hat{B}_i)$ and $Tl(\hat{B}_i)$ as $\hat{T}f_i$ and $\hat{T}l_i$ (predicted binary code), and $Tf(B_i)$ and $Tl(B_i)$ as Tf_i and Tl_i (target binary code) for brevity. Expected value of the absolute error can be further expanded as:

$$\begin{aligned} \mathbb{E}(|\hat{y}_i^{\text{BEL-J}} - y_i|) &= \mathbb{E}(|\hat{T}f_i + \hat{T}l_i + Tc - (Tf_i + Tl_i + Tc)|) \\ &= \mathbb{E}(|(\hat{T}f_i - Tf_i) + (\hat{T}l_i - Tl_i)|) \\ &\leq \mathbb{E}(|\hat{T}f_i - Tf_i| + |\hat{T}l_i - Tl_i|) \\ &= \mathbb{E}(|\hat{T}f_i - Tf_i|) + \mathbb{E}(|\hat{T}l_i - Tl_i|) \end{aligned} \quad (9)$$

Thus, the sum of the expected error of $Tf()$ and $Tl()$ is the upper bound of the label’s expected error. Further, we derive the relation between binary classifiers’ error probabilities and $\mathbb{E}(|\hat{T}f_i - Tf_i|)$ and $\mathbb{E}(|\hat{T}l_i - Tl_i|)$.

We consider $Q_i = n$, where $1 \leq n \leq \frac{N}{2}$ for our derivation. In such a case, $Tf_i = n$ and $Tl_i = -\frac{N}{2}$. However, as the code is symmetric around $Q_i = \frac{N}{2}$, it can be shown that the derived equation for $\mathbb{E}|\hat{y}_i - y_i|$ can be used for $1 \leq Q_i \leq N - 1$.

1. Derivation of $\mathbb{E}|\hat{T}f_i - Tf_i|$: As shown in Equation 8, $Tf()$ finds the location k of the first occurrence of “1” in the binary sequence. In the case of an erroneous binary sequence, the position of the first occurrence of “1” might shift, which results in an error between $\hat{T}f_i$ and Tf_i . Figure 3 shows examples of the correct and erroneous outputs of classifiers for label $Q_i = n$. For label $Q_i = n$, $b_i^k = 0$ for $k \in \{1, 2, \dots, \frac{N}{2} - n\}$ and $b_i^k = 1$ for $k \in \{\frac{N}{2} - n + 1, \frac{N}{2} - n + 2, \dots, \frac{N}{2}\}$.

For case 1, error in a classifier C^k , $k \in \{1, 2, \dots, \frac{N}{2} - n\}$ is considered, where $b_i^k = 0$ and $\hat{b}_i^k = 1$. For $k \in \{1, 2, \dots, \frac{N}{2} - n\}$, an error at classifier C^k will result in erroneous $\hat{T}f_i$ only if all proceeding classifiers are correct, since if any of the proceeding classifier z is incorrect, i.e. $\hat{b}_i^z = 1$, then the location of the first occurrence of “1” will be shifted to z , and any error in the following classifiers

will not affect the value of $\hat{T}f$. Such a case ($\hat{b}_i^1 = T, \hat{b}_i^2 = T, \dots, \hat{b}_i^{k-1} = T, \hat{b}_i^k = F, \hat{b}_i^{k+1} = T/F, \dots, \hat{b}_i^{\frac{N}{2}} = T/F$) considers a total of $2^{\frac{N}{2}-k}$ combinations out of $2^{\frac{N}{2}}$ for $k \in \{1, 2, \dots, \frac{N}{2} - n\}$. Assuming that the binary classifiers are mutually independent, the error value and the probability of this combination can be shown to be:

$$|\hat{T}f_i - Tf_i| = \left(\frac{N}{2} - n + 1 - k\right) \quad (10)$$

$$\begin{aligned} Pr(\hat{b}_i^1 = T, \hat{b}_i^2 = T, \dots, \hat{b}_i^{k-1} = T, \hat{b}_i^k = F) &= Pr(\hat{b}_i^1 = T)Pr(\hat{b}_i^2 = T)\dots Pr(\hat{b}_i^{k-1} = T)Pr(\hat{b}_i^k = F) \\ &= \left(\prod_{j=1}^{k-1} (1 - e_j(n))\right) \cdot e_k(n) \end{aligned} \quad (11)$$

The above term considers combinations ($b'^1 = T, b'^2 = T, \dots, b'^{k-1} = T, b'^k = F, b'^{k+1} = T/F, \dots, b'^{\frac{N}{2}} = T/F$) for $k \in \{1, 2, \dots, \frac{N}{2} - n\}$, which constitutes to a total of $\sum_{k=1}^{\frac{N}{2}-n} 2^{\frac{N}{2}-k}$ combinations out of $2^{\frac{N}{2}}$.

For case 2, error in a classifier $C^k, k \in \{\frac{N}{2} - n + 1, \frac{N}{2} - n + 2, \dots, \frac{N}{2}\}$ is considered, where $b_i^k = 1$ and $\hat{b}_i^k = 0$. We consider a combination ($\hat{b}_i^1 = T, \hat{b}_i^2 = T, \dots, \hat{b}_i^{\frac{N}{2}-n} = T, \hat{b}_i^{\frac{N}{2}-n+1} = F, \dots, \hat{b}_i^{k-1} = F, \hat{b}_i^k = T, \hat{b}_i^{k+1} = T/F, \dots, \hat{b}_i^{\frac{N}{2}} = T/F$). For this case, the position of the first occurrence of “1” will be moved to k , which will result in erroneous $\hat{T}f_i$. Such a case would cover $2^{\frac{N}{2}-k}$ combinations out of $2^{\frac{N}{2}}$ for $k \in \{\frac{N}{2} - n + 1, \frac{N}{2} - n + 2, \dots, \frac{N}{2}\}$. The error value and the probability of this combination can be shown to be:

$$|\hat{T}f_i - Tf_i| = \left(k - \left(\frac{N}{2} - n + 1\right)\right) \quad (12)$$

$$\begin{aligned} Pr(\hat{b}_i^1 = T, \hat{b}_i^2 = T, \dots, \hat{b}_i^{\frac{N}{2}-n} = T, \hat{b}_i^{\frac{N}{2}-n+1} = F, \dots, \hat{b}_i^{k-1} = F, \hat{b}_i^k = T) &= \\ \left(\prod_{j=1}^{\frac{N}{2}-n} (1 - e_j(n))\right) \cdot \left(\prod_{j=\frac{N}{2}-n+1}^{k-1} e_j(n)\right) \cdot (1 - e_k(n)) \end{aligned} \quad (13)$$

The above term considers combinations ($\hat{b}_i^1 = T, \hat{b}_i^2 = T, \dots, \hat{b}_i^{\frac{N}{2}-n} = T, \hat{b}_i^{\frac{N}{2}-n+1} = F, \dots, \hat{b}_i^{k-1} = F, \hat{b}_i^k = T, \hat{b}_i^{k+1} = T/F, \dots, \hat{b}_i^{\frac{N}{2}} = T/F$), which constitutes to a total of $\sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} 2^{\frac{N}{2}-k}$ combinations out of $2^{\frac{N}{2}}$ for $k \in \{\frac{N}{2} - n + 1, \frac{N}{2} - n + 2, \dots, \frac{N}{2}\}$.

Combining Equation 10 to Equation 13, the expected value of $|\hat{T}f_i - Tf_i|$ can be derived as:

$$\begin{aligned} \mathbb{E}(|\hat{T}f_i - Tf_i|) &= \sum_{k=1}^{\frac{N}{2}-n} \left(\frac{N}{2} - n + 1 - k\right) \cdot \left(\prod_{j=1}^{k-1} (1 - e_j(n))\right) \cdot e_k(n) \\ &\quad + \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(k - \left(\frac{N}{2} - n + 1\right)\right) \cdot \left(\prod_{j=1}^{\frac{N}{2}-n} (1 - e_j(n))\right) \cdot \left(\prod_{j=\frac{N}{2}-n+1}^{k-1} e_j(n)\right) \cdot (1 - e_k(n)) \\ &= \sum_{k=1}^{\frac{N}{2}-n} \left(\frac{N}{2} - n + 1 - k\right) \cdot e_k(n) \cdot \left(\prod_{j=1}^{k-1} (1 - e_j(n))\right) + \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(\prod_{j=\frac{N}{2}-n+1}^k e_j(n)\right) \end{aligned} \quad (14)$$

The first term in Equation 14 covers $\sum_{k=1}^{\frac{N}{2}-n} 2^{\frac{N}{2}-k}$ combinations and the second term considers $\sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} 2^{\frac{N}{2}-k}$ combinations. Adding one combination where all the classifiers are correct, Equation 14 considers all of the possible combinations $2^{\frac{N}{2}}$ to find expected value of $|\hat{T}f_i - Tf_i|$.

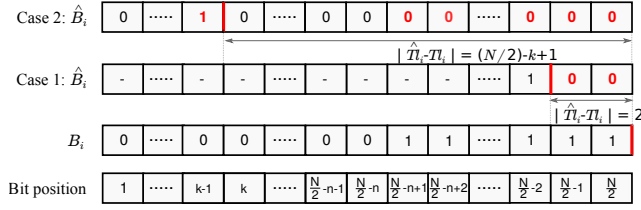


Figure 4: Effect of classifier error on $\hat{T}l_i - Tl_i$ for label $Q_i = n$. Case 1 and case 2 represent erroneous outputs. 0/1 highlighted in red color represents an error in the classifier's output. "-" represents error/no error in both cases.

2. Derivation of $\mathbb{E}|\hat{T}l_i - Tl_i|$: As shown in Equation 8, $Tl_i()$ finds the location k of the last occurrence of "1" in the binary sequence. In the case of an erroneous binary sequence, the position of the last occurrence of "1" might shift, which results in an erroneous value of $\hat{T}l_i$. Figure 4 shows examples of correct and erroneous outputs of classifiers for label $Q_i = n$.

For case 1, an error in a classifier C^k , $k \in \{\frac{N}{2} - n + 1, \frac{N}{2} - n + 2, \dots, \frac{N}{2}\}$ is considered, where $b_i^k = 1$ and $\hat{b}_i^k = 0$. We consider a combination ($\hat{b}_i^{\frac{N}{2}} = F, \hat{b}_i^{\frac{N}{2}-1} = F, \dots, \hat{b}_i^{k+1} = F, \hat{b}_i^k = T, \hat{b}_i^{k-1} = T/F, \dots, \hat{b}_i^1 = T/F$). For this case, position of the last occurrence of "1" will be moved to k , which will result in erroneous $\hat{T}l_i$. Such a case would cover 2^{k-1} combinations out of $2^{\frac{N}{2}}$. The error value and the probability of this combination can be shown to be:

$$|\hat{T}l_i - Tl_i| = \left(\frac{N}{2} - k\right) \quad (15)$$

$$Pr(\hat{b}_i^{\frac{N}{2}} = F, \hat{b}_i^{\frac{N}{2}-1} = F, \dots, \hat{b}_i^{k+1} = F, \hat{b}_i^k = T) = \left(\prod_{j=k+1}^{\frac{N}{2}} e_j(n)\right) \cdot (1 - e_k(n)) \quad (16)$$

The above term considers combinations ($\hat{b}_i^{\frac{N}{2}} = F, \hat{b}_i^{\frac{N}{2}-1} = F, \dots, \hat{b}_i^{k+1} = F, \hat{b}_i^k = T, \hat{b}_i^{k-1} = T/F, \dots, \hat{b}_i^1 = T/F$) for $k \in \{\frac{N}{2} - n + 1, \frac{N}{2} - n + 2, \dots, \frac{N}{2}\}$, which constitutes to a total of $\sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} 2^{k-1}$ combinations out of $2^{\frac{N}{2}}$.

For case 2, an error in a classifier C^k , $k \in \{1, 2, \dots, \frac{N}{2} - n\}$ is considered, where $b_i^k = 0$ and $\hat{b}_i^k = 1$. We consider a combination ($\hat{b}_i^{\frac{N}{2}} = F, \dots, \hat{b}_i^{\frac{N}{2}-n+1} = F, \hat{b}_i^{\frac{N}{2}-n} = T, \dots, \hat{b}_i^{k+1} = T, \hat{b}_i^k = F, \hat{b}_i^{k-1} = T/F, \dots, \hat{b}_i^1 = T/F$). For this case, position of the last occurrence of "1" will be moved to k , which will result in erroneous $\hat{T}l_i$. Such a case would cover 2^{k-1} combinations out of $2^{\frac{N}{2}}$. The error value and the probability of this combination can be shown to be:

$$|\hat{T}l_i - Tl_i| = \left(\frac{N}{2} - k\right) \quad (17)$$

$$Pr(\hat{b}_i^{\frac{N}{2}} = F, \dots, \hat{b}_i^{\frac{N}{2}-n+1} = F, \hat{b}_i^{\frac{N}{2}-n} = T, \dots, \hat{b}_i^{k+1} = T, \hat{b}_i^k = F) = \left(\prod_{j=\frac{N}{2}-n+1}^{\frac{N}{2}} e_j(n)\right) \cdot \left(\prod_{j=k+1}^{\frac{N}{2}-n} (1 - e_j(n))\right) \cdot (e_k(n)) \quad (18)$$

The above term considers combinations ($\hat{b}_i^{\frac{N}{2}} = F, \dots, \hat{b}_i^{\frac{N}{2}-n+1} = F, \hat{b}_i^{\frac{N}{2}-n} = T, \dots, \hat{b}_i^{k+1} = T, \hat{b}_i^k = F, \hat{b}_i^{k-1} = T/F, \dots, \hat{b}_i^1 = T/F$) for $k \in \{1, 2, \dots, \frac{N}{2} - n\}$, which constitutes to a total of $\sum_{k=1}^{\frac{N}{2}-n} 2^{k-1}$ combinations out of $2^{\frac{N}{2}}$.

Combining Equation 15 to Equation 18, the expected value of $|\hat{T}l_i - Tl_i|$ can be derived

as:

$$\begin{aligned}
\mathbb{E}(|\hat{T}l_i - Tl_i|) &= \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(\frac{N}{2} - k\right) \cdot \left(\prod_{j=k+1}^{\frac{N}{2}} e_j(n)\right) \cdot (1 - e_k(n)) \\
&\quad + \sum_{k=1}^{\frac{N}{2}-n} \left(\frac{N}{2} - k\right) \cdot \left(\prod_{j=\frac{N}{2}-n+1}^{\frac{N}{2}} e_j(n)\right) \cdot \left(\prod_{j=k+1}^{\frac{N}{2}-n} (1 - e_j(n))\right) \cdot (e_k(n)) \quad (19) \\
&= \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(\prod_{j=k}^{\frac{N}{2}} e_j(n)\right) + \left(\prod_{j=\frac{N}{2}-n+1}^{\frac{N}{2}} e_j(n)\right) \cdot \sum_{k=1}^{\frac{N}{2}-n} \left(\prod_{j=k}^{\frac{N}{2}-n} (1 - e_j(n))\right)
\end{aligned}$$

The first term in Equation 19 covers $\sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} 2^{k-1}$ combinations and the second term considers $\sum_{k=1}^{\frac{N}{2}-n} 2^{k-1}$ combinations. Adding one combination where all the classifiers are correct, Equation 19 considers all of the possible combinations $2^{\frac{N}{2}}$ to find expected value of $|\hat{T}l_i - Tl_i|$.

Combining Equation 9, Equation 14, and Equation 19, the expected value of error for $Q_i = n$ in terms of classifiers' error probabilities can be derived as:

$$\begin{aligned}
\mathbb{E}(\hat{y}_i^{\text{BEL-J}} - y_i) &\leq \sum_{k=1}^{\frac{N}{2}-n} \left(\frac{N}{2} - n + 1 - k\right) \cdot e_k(n) \cdot \left(\prod_{j=1}^{k-1} (1 - e_j(n))\right) + \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(\prod_{j=\frac{N}{2}-n+1}^k e_j(n)\right) \\
&\quad + \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(\prod_{j=k}^{\frac{N}{2}} e_j(n)\right) + \left(\prod_{j=\frac{N}{2}-n+1}^{\frac{N}{2}} e_j(n)\right) \cdot \sum_{k=1}^{\frac{N}{2}-n} \left(\prod_{j=k}^{\frac{N}{2}-n} (1 - e_j(n))\right) \quad (20)
\end{aligned}$$

As the binary code is symmetric around $\frac{N}{2}$ as shown in Figure 2, the expected errors for label $y_i \in [1, \frac{N}{2}]$ can be mirrored to find expected errors for label $y_i \in [\frac{N}{2}, N-1]$. For a uniform distribution of target labels in the range $[1, N-1]$, the expected error can be derived as:

$$\begin{aligned}
\mathbb{E}(\hat{y}^{\text{BEL-J}} - y) &\leq \frac{1}{N-1} \sum_{n=1}^{N-1} \left[\sum_{k=1}^{\frac{N}{2}-n} \left(\frac{N}{2} - n + 1 - k\right) \cdot e_k(n) \cdot \left(\prod_{j=1}^{k-1} (1 - e_j(n))\right) + \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(\prod_{j=\frac{N}{2}-n+1}^k e_j(n)\right) \right. \\
&\quad \left. + \sum_{k=\frac{N}{2}-n+1}^{\frac{N}{2}} \left(\prod_{j=k}^{\frac{N}{2}} e_j(n)\right) + \left(\prod_{j=\frac{N}{2}-n+1}^{\frac{N}{2}} e_j(n)\right) \cdot \sum_{k=1}^{\frac{N}{2}-n} \left(\prod_{j=k}^{\frac{N}{2}-n} (1 - e_j(n))\right) \right] \quad (21)
\end{aligned}$$

We also verify the equation by comparing the expected value of error based on Equation 20 for $Q_i \in \{1, 2, \dots, N-1\}$ with the expected error calculated by 100,000 random samples of binary sequences for the same error probabilities $e_k(n)$. Figure 5 compares the expected error from Equation 20 and measured from statistical samples, and validates error upper bounds calculated using Equation 20 and Equation 21.

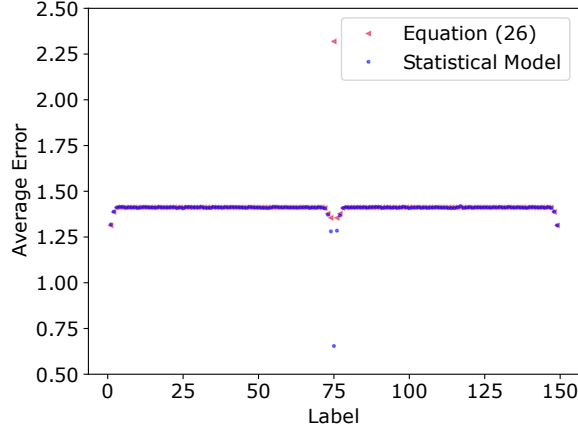


Figure 5: Comparison of expected value of error from Equation 20 and random samples for given error probabilities of the classifiers.

C ERROR PROBABILITY OF CLASSIFIERS

It is known that the error/misclassification probability $e_k(n)$ of a classifier tends to increase as the target label value n is closer to the classifier’s decision boundaries (Cardoso & Pinto da Costa, 2007). We approximate $e_k(y)$ for a classifier C^k with t bit transitions as a linear combination of t Gaussian distributions. Here, each Gaussian term is centered around a bit transition. Figure 6 shows the empirically observed error probability distributions for different classifiers trained for different combinations of network and dataset. We also show the approximate error probability distribution using a linear combination of Gaussian distributions. Here r is a scalar multiplied with probability density of gaussian distribution and σ is the standard deviation (Equation 3 and 4).

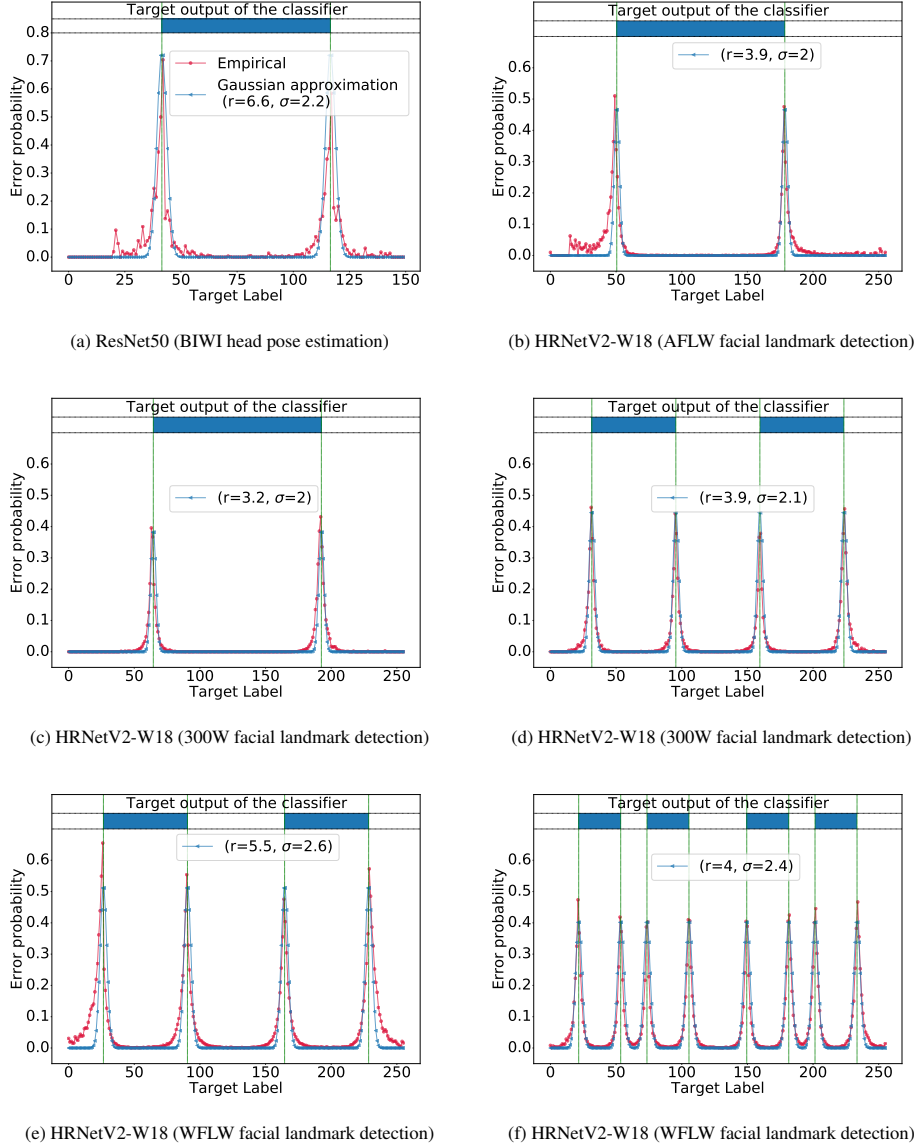


Figure 6: Classification error probability versus target label y for different classifiers. The top horizontal bar represent target output of the classifier. Blue color represents output 1.

D EXPERIMENTAL METHODOLOGY

All experiments are conducted on a Linux machine with an Intel i9-9900X processor and an Nvidia RTX 2080 Ti GPU with 11GB of memory. Our code is implemented using Python 3.8.3 with Pytorch 1.5.1 using CUDA 10.2. Our evaluation is averaged over 5 training runs with separate seeds.

D.1 HEAD POSE ESTIMATION

Head pose estimation aims to find a human head’s pose in terms of three angles: yaw, pitch, and roll. In this work, we consider landmark-free 2D head pose estimation.

Datasets: We follow the evaluation setting of Hopenet (Ruiz et al., 2018) and FSA-Net (fsa, 2019) and use two evaluation protocols with three widely used datasets: 300W-LP (Zhu et al., 2016), BIWI (Fanelli et al., 2013), and AFLW2000 (Zhu et al., 2016).

Protocol 1: BIWI dataset is used for training and evaluation in this protocol. BIWI dataset consists of 24 videos of 20 subjects with total 15,128 frames. Three random splits of 70%-30% images are used for training and evaluation. For the BIWI dataset, the yaw angle is in the range $[-75^\circ, 75^\circ]$, the pitch is in the range $[-65^\circ, 85^\circ]$, and the roll angle is in the range $[-55^\circ, 45^\circ]$.

Protocol 2: In this setting, the synthetic 300W-LP dataset is used for training, consisting of 122,450 samples. The trained network is tested on a real-world AFLW2000 dataset. Yaw, pitch, and roll angles are in the range $[-99^\circ, 99^\circ]$ for both datasets.

Evaluation metrics: Mean Absolute Error (MAE) between the target and predicted values is used as the evaluation metric for this benchmark. MAE for a regression task is defined as:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N \frac{1}{P} \sum_{j=1}^P |y_{i,j} - \hat{y}_{i,j}| \quad (22)$$

Here, N is the number of test samples, and P is the dimension of the regression task output. For head pose estimation, the dimension of regression output is three (i.e., yaw, pitch, and roll). y is the target, and \hat{y} is the predicted label.

Network architecture and training parameters: We evaluate our approach on two models: ResNet-50 and RAFA-Net. With ResNet-50, two runs with different random seeds for each combination of learning rate $\{0.001, 0.0001, 0.00001\}$ and batch size $\{8, 16\}$ are used for hyperparameter tuning. For data augmentation, images are loosely cropped around the center in the training and testing datasets with random flipping. With RAFA-Net, we use the training parameters and data augmentation used in Behera et al. (2021).

We refer to Protocol 1 evaluated with ResNet-50 as **HPE1**, Protocol 1 evaluated with RAFA-Net as **HPE3**, Protocol 2 evaluated with ResNet-50 as **HPE2**, and Protocol 2 evaluated with RAFA-Net as **HPE4**. Table 19 provides a summary of the training parameters used with protocol 1. Table 20 provides a summary of the training parameters used with protocol 2.

Table 19: Training parameters for head pose estimation with protocol 1.

Approach	Label range/Quantization levels	Optimizer	Epochs	Batch size	Learning rate	Learning rate schedule	Training time (GPU hours)
HPE1	Yaw: $[-75^\circ, 75^\circ]/150$, Pitch: $[-65^\circ, 85^\circ]/150$, Roll: $[-55^\circ, 45^\circ]/100$	Adam, weight decay=0, momentum = 0	50	8	0.0001	1/10 after 30 Epochs	2
HPE3	$[-179^\circ, 180^\circ]/360$	RMSProp, momentum=0, rho = 0.9	100	16	0.001	-	6

Table 20: Training parameters for head pose estimation with protocol 2.

Approach	Label range/Quantization levels	Optimizer	Epochs	Batch size	Learning rate	Learning rate schedule	Training time (GPU hours)
HPE2	$[-99^\circ, 99^\circ]/200$	Adam, weight decay=0, momentum = 0	20	16	0.00001	1/10 after 10 Epochs	4
HPE4	$[-179^\circ, 180^\circ]/360$	RMSProp, momentum=0, rho = 0.9	100	16	0.001	-	48

Related work Existing approaches for head pose estimation include stage-wise soft regression (Yang et al., 2018; fsa, 2019), a combination of classification and regression (Mukherjee & Robertson, 2015; Ruiz et al., 2018), and ordinal regression (Hsu et al., 2019). SSR-Net (Yang et al.,

2018) proposes the use of stage-wise soft regression to use the softmax values of classification output to refine the label. FSA-Net (fsa, 2019) proposes extending stage-wise estimation to head pose estimation using feature aggregation. HopeNet (Ruiz et al., 2018) uses a combination of classification and regression loss to train a model for head pose estimation. Whereas, QuatNet (Hsu et al., 2019) proposes a combination of L2 loss and a custom ordinal regression loss. RAFA-Net (Behera et al., 2021) uses an attention based approach for feature extraction with direct regression.

We compare BEL with the performance of related work in Table 21 and Table 22. 95% confidence intervals are given.

Table 21: Landmark-free 2D Head poses estimation evaluation for protocol 1 (HPE1 and HPE3).

Approach	Feature Extractor	#Params (M)	Yaw	Pitch	Roll	MAE
SSR-Net-MD (Yang et al., 2018) (Soft regression)	SSR-Net	1.1	4.24	4.35	4.19	4.26
FSA-Caps-Fusion (fsa, 2019) (Soft regression)	FSA-Net	5.1	<u>2.89</u>	4.29	3.60	3.60
Direct regression (L2 loss)	ResNet50 (HPE1)	23.5	4.62	5.24	4.43	4.76 \pm 0.35
BEL-U/GEN-EX/L2	ResNet50 (HPE1)	23.6	3.32	3.80	3.53	3.56 \pm 0.01
RAFA-Net (Behera et al., 2021) (Direct Regression)	RAFA-Net (HPE3)	69.8	3.07	4.30	<u>2.82</u>	3.40
BEL-B1JDJ/GEN-EX/BCE	RAFA-Net (HPE3)	69.8	3.21	3.34	3.43	3.30 \pm 0.04

Table 22: Landmark-free 2D Head poses estimation evaluation for protocol 2 (HPE2 and HPE4).

Approach	Feature Extractor	#Params (M)	Yaw	Pitch	Roll	MAE
SSR-Net-MD (Yang et al., 2018) (Soft regression)	SSR-Net	1.1	5.14	7.09	5.89	6.01
FSA-Caps-Fusion (fsa, 2019) (Soft regression)	FSA-Net	5.1	4.50	6.08	4.64	5.07
HopeNet* ($\alpha = 2$) (Ruiz et al., 2018) (classification + regression loss)	ResNet50	23.9	6.47	6.56	5.44	6.16
Direct regression (L2 loss)	ResNet50 (HPE2)	23.5	5.85	6.34	4.80	5.65 \pm 0.13
BEL-U/GEN-EX/BCE	ResNet50 (HPE2)	23.6	4.54	5.76	3.96	4.77 \pm 0.05
RAFA-Net (Behera et al., 2021) (Direct Regression)	RAFA-Net (HPE4)	69.8	3.60	4.92	3.88	4.13
BEL-U/GEN-EX/BCE	RAFA-Net (HPE4)	69.8	<u>3.28</u>	<u>4.78</u>	<u>3.55</u>	<u>3.90</u> \pm 0.03

D.2 FACIAL LANDMARK DETECTION

Facial landmark detection is a problem of detecting the (x, y) coordinates of keypoints in a given face image.

Datasets We use the COFW (Burgos-Artizzu et al., 2013), 300W (Sagonas et al., 2013), WFLW (Wu et al., 2018), and AFLW (Köstinger et al., 2011) datasets with data augmentation and evaluation protocols described in (Wang et al., 2020). Data augmentation is performed by random flipping, 0.75 – 1.25 scaling, and ± 30 degrees in-plane rotation for all the datasets. We use 256 quantization levels for binary-encoded labels.

COFW: The COFW dataset (Burgos-Artizzu et al., 2013) consists of 1, 345 training and 507 testing images. Each image is annotated with 29 facial landmarks.

300W: This dataset is a combination of HELEN, LFPW, AFW, XM2VTS, and IBUG datasets. Each image is annotated with 68 facial landmarks. The training dataset consists of 3, 148 images. We evaluate the trained model on four test sets: full test set with 689 images, common subset with 554

images from HELEN and LFPW, challenging subset with 135 images from IBUG, and the official test set with 300 indoor and 300 outdoor images.

WFLW: WFLW dataset consists of 7,500 training images where each image is annotated with 98 facial landmarks. Full test dataset consists of 2,500 images. We use test subsets: large pose (326 images), expression (314 images), illumination (698 images), make-up (206 images), occlusion (736 images), and blur (773 images).

AFLW: Each image has 19 annotated facial key points in this dataset. AFLW dataset consists of 20,000 training images where each image is annotated with 19 facial landmarks. The full test dataset consists of 4,836 images, and the frontal test set consists of 1,314 images.

Evaluation metrics: Mean Normalized Error (NME) between the target and predicted values is used as the evaluation metric for this benchmark. NME for a regression task is defined as:

$$\text{NME} = \frac{1}{N} \sum_{i=1}^N \frac{1}{P} \cdot \frac{1}{L} \sum_{j=1}^P |y_{i,j} - \hat{y}_{i,j}|_2 \quad (23)$$

Here, N is the number of test samples, and P is the dimension of the regression task output, i.e., the number of landmarks for facial landmark detection. y is the target, and \hat{y} is the predicted label. L is the normalization factor. Inter-ocular distance normalization is used for COFW, 300W, and WFLW datasets, and bounding box-based normalization is used for AFLW dataset.

We also report failure rate (f@10%) for some datasets. The failure rate (f@10%) is defined as the fraction of test samples with normalized errors higher than 0.1.

Network architecture and training parameters: We evaluate BEL by applying it on HRNetV2-W18. HRNetV2-W18 feature extractor’s output is 240 channels of size 64×64 . For heatmap regression, a 1×1 convolution is used to get P heatmaps of size 64×64 , where P is the number of landmarks. Since BEL-x predicts (x, y) coordinates directly we modify the architecture of HRNetV2-W18 to support direct prediction of landmarks. Figure 7 shows the modified architecture of HRNetV2-W18 for BEL-x.

The state-of-the-art approaches for facial landmark detection uses heatmap regression, which minimizes the pixel-level loss between the predicted and target heatmaps. We evaluate the applicability of BEL on heatmap regression in Appendix A. In contrast, BEL-x predicts (x, y) coordinates directly with 256 quantization levels.

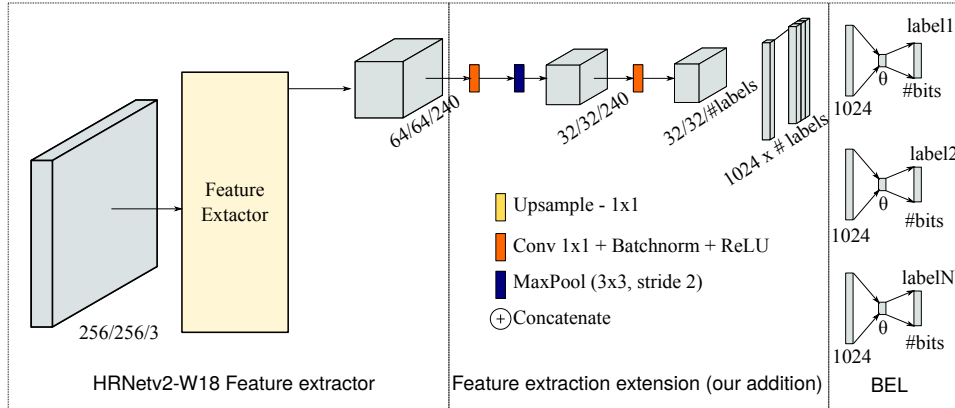


Figure 7: HRNetV2-W18 feature extractor combined with BEL regressor for (x, y) coordinates

We use two runs with different random seeds to decide the learning rate. We consider learning rates $\{0.0003, 0.0005, 0.0007\}$ and $\theta \in \{10, 30\}$.

Table 23 provides a summary of all the training parameters. We refer to HRNetV2-W18 evaluated on COFW as **FLD1**, on 300W as **FLD2**, on WFLW as **FLD3**, and on AFLW as **FLD4**.

Table 23: Training parameters for facial landmark detection for HRNetV2-W18 feature extractor.

Dataset	Optimizer	Epochs	Batch size	Learning rate (BEL/Direct regression/Multiclass classification)	Learning rate schedule	Training time (GPU hours)
COFW	Adam, weight decay=0, momentum = 0	60	8	0.0005/0.0003/ 0.0003	1/10 after 30 and 50 Epochs	$\frac{1}{2}$
300W	Adam, weight decay=0, momentum = 0	60	8	0.0007/0.0003/ 0.0003	1/10 after 30 and 50 Epochs	3
WFLW	Adam, weight decay=0, momentum = 0	60	8	0.0003/0.0003/ 0.0003	1/10 after 30 and 50 Epochs	5
AFLW	Adam, weight decay=0, momentum = 0	60	8	0.0005/0.0005/ 0.0003	1/10 after 30 and 50 Epochs	8

Related work Facial landmark detection is an extensively studied problem used for facial analysis and modeling. Common regression approaches for this tasks includes regression using MSE loss (Xiong & De la Torre, 2013; Lv et al., 2017), cascaded regression (Miao et al., 2018; Tzimiropoulos, 2015; Zhu et al., 2016; Sun et al., 2013), and coarse-to-fine regression (Sun et al., 2013; Shizhan Zhu et al., 2015; Zhang et al., 2014). State-of-the-art methods for this task learn heatmaps by regression to find facial landmarks. SAN (Dong et al., 2018) augments training data using temporal information and GAN-generated faces. DVLN (Wu & Yang, 2017), CFSS (Shizhan Zhu et al., 2015), LAB (Wu et al., 2018), DSRN (Miao et al., 2018) take advantage of correlations between facial landmarks. DAN (Kowalski et al., 2017) introduces a progressive refinement approach using predicted landmark heatmaps. LAB (Wu et al., 2018) also exploits extra boundary information to improve the accuracy. LUVLi (Kumar et al., 2020) proposes a landmark’s location, uncertainty, and visibility likelihood-based loss. Bulat & Tzimiropoulos (2016) proposes the use of binary heatmaps with pixel-wise binary cross-entropy loss. AWing (Wang et al., 2019) proposes adapted wing loss to improve the accuracy of heatmap regression. AnchorFace (Xu et al., 2020) demonstrates that anchoring facial landmarks on templates improves regression performance for large poses. HRNet (Wang et al., 2020) proposes a CNN architecture to maintain high-resolution representations across the network, and uses heatmap regression. The target heatmap is generated by assuming a Gaussian distribution around the landmark location.

We compare BEL with related work in Table 24- 27. 95% confidence intervals are provided.

Table 24: Facial landmark detection results on COFW dataset (FLD1). The failure rate is measured at the threshold 0.1. $\theta = 30$ is used for BEL.

Approach	Feature Extractor	#Params/ GFlops	Test NME	FR _{0.1}
LAB (w B) (Wu et al., 2018)	Hourglass	25.1/19.1	3.92	0.39
AWing (Wang et al., 2019)*	Hourglass	25.1/19.1	4.94	-
HRNetV2-W18 (Wang et al., 2020) (Heatmap regression)	HRNetV2-W18	9.6/4.6	3.45	0.19
Direct regression (L2 loss)	HRNetV2-W18	10.2/4.7	3.96 ± 0.02	0.29
Direct regression (L1 loss)	HRNetV2-W18	10.2/4.7	3.60 ± 0.02	0.29
BEL-HEXJ/GEN-EX/CE	HRNetV2-W18	10.6/4.6	$\underline{3.34} \pm 0.02$	0.40

*Uses different data augmentation for the training

Table 25: Facial landmark detection results on 300W dataset (FLD2). $\theta = 10$ is used for BEL.

Approach	Feature Extractor	#Params/ GFlops	Test	Common	Challenging	Full
DAN (Kowalski et al., 2017)	-	-	-	3.19	5.24	3.59
LAB (w B) (Wu et al., 2018)	Hourglass	25.1/19.1	-	2.98	5.19	3.49
AnchorFace (Xu et al., 2020)	ShuffleNet-V2	-	-	3.12	6.19	3.72
AWing (Wang et al., 2019)*	Hourglass	25.1/19.1	-	2.72	4.52	3.07
LUVLi (Kumar et al., 2020)	CU-Net	-	-	2.76	5.16	3.23
HRNetV2-W18 (Wang et al., 2020) (Heatmap regression)	HRNetV2-W18	9.6/4.6	-	2.87	5.15	3.32
Direct regression (L2 loss)	HRNetV2-W18	10.2/4.7	4.40	3.25	5.65	3.71 \pm 0.05
Direct regression (L1 loss)	HRNetV2-W18	10.2/4.7	4.26	3.10	5.42	3.54 \pm 0.03
BEL-U/GEN-EX/CE	HRNetV2-W18	11.2/4.6	4.09	2.91	5.50	3.40 \pm 0.02

*Uses different data augmentation for the training

Table 26: Facial landmark detection results (NME) on WFLW test (FLD3) and 6 subsets: pose, expression (expr.), illumination (illu.), make-up (mu.), occlusion (occu.) and blur. $\theta = 10$ is used for BEL.

Approach	Feature Extractor	#Params/ GFlops	Test	Pose	Expr.	Illu.	MU	Occu.	Blur
LAB (w B) (Wu et al., 2018)	Hourglass	25.1/19.1	5.27	10.24	5.51	5.23	5.15	6.79	6.32
AnchorFace (Xu et al., 2020)*	HRNetV2-W18	-/5.3	<u>4.32</u>	7.51	4.69	<u>4.20</u>	4.11	<u>4.98</u>	<u>4.82</u>
AWing (Wang et al., 2019)*	Hourglass	25.1/19.1	4.36	<u>7.38</u>	<u>4.58</u>	4.32	4.27	5.19	4.96
LUVLi (Kumar et al., 2020)	CU-Net	-	4.37	-	-	-	-	-	-
HRNetV2-W18 (Wang et al., 2020) (Heatmap regression)	HRNetV2-W18	9.6/4.6	4.60	7.94	4.85	4.55	4.29	5.44	5.42
Direct regression (L2 loss)	HRNetV2-W18	10.2/4.7	5.56 \pm 0.05	10.17	6.13	5.49	5.29	6.83	6.52
Direct regression (L1 loss)	HRNetV2-W18	10.2/4.7	4.64 \pm 0.03	8.13	4.96	4.49	4.45	5.41	5.25
BEL-B1JDJ/GEN-EX/CE	HRNetV2-W18	11.7/4.6	4.36 \pm 0.02	7.53	4.64	4.28	4.19	5.19	5.05

*Uses different data augmentation for the training

Table 27: Facial landmark detection results on AFLW dataset (FLD4). $\theta = 30$ is used for BEL.

Approach	Feature Extractor	#Params/ GFlops	Full	Frontal
LAB (w/o B) (Wu et al., 2018)	Hourglass	25.1/19.1	1.85	1.62
AnchorFace (Xu et al., 2020)	ShuffleNet-V2	-	1.56	
LUVLi (Kumar et al., 2020)	CU-Net	-	<u>1.39</u>	<u>1.19</u>
HRNetV2-W18 (Wang et al., 2020) (Heatmap regression)	HRNetV2-W18	9.6/4.6	1.57	1.46
Direct regression (L2 loss)	HRNetV2-W18	10.2/4.7	2.10 \pm 0.02	1.71
Direct regression (L1 loss)	HRNetV2-W18	10.2/4.7	1.51 \pm 0.01	1.34
BEL-B1JDJ/GEN-EX/CE	HRNetV2-W18	10.8/4.6	1.47 \pm 0.00	1.30

*Uses different data augmentation for the training

D.3 AGE ESTIMATION

Age estimation aims to predict the age given an image of a human head.

Datasets We use the MORPH-II (Ricanek & Tesafaye, 2006) and AFAD (Niu et al., 2016) datasets for our evaluation. Cumulative Score (CS) and MAE are used as evaluation metrics. We preprocess the MORPH-II dataset by aligning images first along the average eye position (Raschka, 2018), then by re-aligning so that the tip of the nose is in the center of each image. We do not preprocess the AFAD dataset as faces are already centered. Afterwards, face images are resized to $256 \times 256 \times 3$ and randomly cropped to $224 \times 224 \times 3$ for training. For testing, a center crop of $224 \times 224 \times 3$ is taken.

MORPH-II: This dataset consists of 55,608 face images with age labels between 16 and 70. The dataset is randomly divided into 39,617 training, 4,398 validation, and 11,001 testing images.

AFAD: This dataset consists of 164,432 Asian facial images and age labels between 15 and 40. The dataset is randomly divided into 118,492 training, 13,166 validation, and 32,763 testing images.

Evaluation metrics: MAE (Equation 22) is used as the evaluation metric. We report Cumulative Score (CS_θ) for some datasets. CS_θ is defined as the fraction of test images with absolute error less than θ years.

Network architecture and training parameters: We evaluate our approach on ResNet-50. We perform two runs with different random seeds to determine the learning rate between $[0.00001, 0.0001, 0.001]$ and use a batch size of 64 for all experiments. We use ImageNet pretrained weights to initialize the network. Full training parameters are described in Table 28. We refer to our evaluation on MORPH-II as **AE1** and AFAD as **AE2**.

Table 28: Training parameters for age estimation using MORPH-II and AFAD dataset

Optimizer	Epochs	Batch size	Learning rate	Learning rate schedule
Adam, weight decay=0, momentum=0	50	64	0.0001	-

Related work Existing approaches for age estimation include ordinal regression (Niu et al., 2016; Cao et al., 2020), soft regression (Yang et al., 2018), and expected value ordinal regression (Pan et al., 2018; Gao et al., 2018). OR-CNN (Niu et al., 2016) proposed the use of ordinal regression via binary classification to predict the label. CORAL-CNN (Cao et al., 2020) refined this approach by enforcing the ordinality of the model output. SSR-Net (Yang et al., 2018) proposed the use of stage-wise soft regression using the softmax of the classification output to refine the predicted label. MV-Loss (Pan et al., 2018) extended the soft regression approach by penalizing the output of the model based on the variance of the age distribution, while DLDL (Gao et al., 2018) proposed to use the KL-divergence between the softmax output and a generated label distribution to train a model.

We compare BEL with related work in Table 29 and Table 30. 95% confidence intervals are provided.

D.4 END-TO-END SELF DRIVING

We evaluate our approach on the NVIDIA PilotNet dataset and PilotNet model for end-to-end autonomous driving (Bojarski et al., 2016). In this task, the steering wheel’s next angle is predicted from an image of the road. We refer to these experiments as **PN**. MAE (Equation 22) is used as the evaluation metric.

Dataset We use a driving dataset consisting of 45,500 images taken around Rancho Palos Verdes and San Pedro, California (Chen). We crop images to $256 \times 70 \times 3$ then resize them to $200 \times 66 \times 3$. We randomly vary the brightness of the image between $[0.2 \times, 1.5 \times]$, randomly flip images, and make random minor perturbations on the steering direction. We use $\theta = 10$ with 670 quantization levels for BEL.

Table 29: Age estimation results on MORPH-II dataset (AE1). $\theta = 10$ is used for BEL.

Approach	Feature extractor	#Parameters (M)	MORPH-II (MAE)	MORPH-II (CS $\theta = 5$)
OR-CNN (Niu et al., 2016) (Ordinal regression by binary classification)	-	1.0	2.58	0.71
MV Loss (Pan et al., 2018) (Direct regression)	VGG-16	138.4	2.41	0.889
DLDL-v2 (Gao et al., 2018) (Ordinal regression with multi-class classification)	ThinAgeNet	3.7	<u>1.96*</u>	-
CORAL-CNN (Cao et al., 2020) (Ordinal regression by binary classification)	ResNet34	21.3	2.49	-
Direct Regression (L2 Loss)	ResNet50	23.1	2.44 ± 0.01	0.903 ± 0.002
BEL-J/BEL-J/BCE	ResNet50	23.1	2.27 ± 0.01	0.928 ± 0.001

*Uses different data augmentation for the training

Table 30: Age estimation results on AFAD dataset (AE2). $\theta = 10$ is used for BEL.

Approach	Feature extractor	#Parameters (M)	AFAD (MAE)	AFAD (CS $\theta = 5$)
OR-CNN (Niu et al., 2016) (Ordinal regression by binary classification)	-	1.0	3.51	0.74
CORAL-CNN (Cao et al., 2020) (Ordinal regression by binary classification)	ResNet34	21.3	3.47	-
Direct Regression (L2 Loss)	ResNet50	23.1	3.21 ± 0.02	0.810 ± 0.02
BEL-B1JDJ/GEN-EX/L1	ResNet50	23.1	3.11 ± 0.01	0.823 ± 0.001

Training parameters We perform two runs with different random seeds to determine the learning rate between $[0.00001, 0.0001, 0.001]$ and use a batch size of 64 for all experiments. Full training parameters are described in Table 31.

Table 31: Training parameters for end-to-end autonomous driving using PilotNet.

Optimizer	Epochs	Batch size	Learning rate	Learning rate schedule
SGD with weight decay=1e-5, momentum=0	50	64	0.1	1/10 at 10, 30 epochs

Related work End-to-end autonomous driving is a novel task that has become increasingly relevant due to the rise of self-driving vehicles. The autonomous driving model’s task is to predict the future driving angle based on a forward-facing image from the perspective of the vehicle. PilotNet (Bojarski et al., 2017) used a small, application-specific network to provide good accuracy within the time constraints of autonomous driving.

We compare BEL with the baseline PilotNet architecture in Table 32. 95% confidence intervals are provided.

REFERENCES

- Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:1087–1096, 2019.
- Ardhendu Behera, Zachary Wharton, Pradeep Hewage, and Swagat Kumar. Rotation axis focused attention network (rafa-net) for estimating head pose. In *Computer Vision – ACCV 2020*, 2021.
- Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *arXiv:1604.07316*, 2016.

Table 32: End-to-end autonomous driving results on PilotNet dataset (PN) and architecture (Bojarski et al., 2017; 2016).

Approach	Feature extractor	#Parameters (M)	MAE
PilotNet (Bojarski et al., 2017)	PilotNet	1.8	4.24 ± 0.45
BEL-J/GEN/CE	PilotNet	1.8	3.11 ± 0.01

Mariusz Bojarski, Philip Yeres, Anna Choromanaska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv:1704.07911*, 2017.

Adrian Bulat and Georgios Tzimiropoulos. Human Pose Estimation via Convolutional Part Heatmap Regression. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 717–732, 2016.

Xavier Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust Face Landmark Estimation under Occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1513–1520, 12 2013. doi: 10.1109/ICCV.2013.191.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331, 2020. doi: <https://doi.org/10.1016/j.patrec.2020.11.008>.

Jaime S. Cardoso and Joaquim F. Pinto da Costa. Learning to Classify Ordinal Data: The Data Replication Method. *Journal of Machine Learning Research*, 8:1393–1429, 2007.

Sully Chen. Driving-datasets. <https://github.com/SullyChen/driving-datasets>.

Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Style Aggregated Network for Facial Landmark Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 379–388, 2018.

Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Gool. Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101(3):437–458, February 2013.

Bin-Bin Gao, Hong-Yu Zhou, Jianxin Wu, and Xin Geng. Age estimation using expectation of label distribution learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 712–718, 7 2018. doi: 10.24963/ijcai.2018/99.

Heng-Wei Hsu, Tung-Yu Wu, Sheng Wan, Wing Hung Wong, and Chen-Yi Lee. Quatnet: Quaternion-based head pose estimation with multiregression loss. *IEEE Transactions on Multimedia*, 21(4): 1035–1046, 2019. doi: 10.1109/TMM.2018.2866770.

Marek Kowalski, Jacek Naruniec, and Tomasz Trzcinski. Deep alignment network: A convolutional neural network for robust face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2034–2043, 2017. doi: 10.1109/CVPRW.2017.254.

Abhinav Kumar, Tim K. Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng. LUVLi face alignment: Estimating Landmarks’ location, uncertainty, and visibility likelihood. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. doi: 10.1109/CVPR42600.2020.00826.

M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2144–2151, 2011.

J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3691–3700, 2017. doi: 10.1109/CVPR.2017.393.

- X. Miao, X. Zhen, X. Liu, C. Deng, V. Athitsos, and H. Huang. Direct shape regression networks for end-to-end face alignment. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5040–5049, 2018.
- S. S. Mukherjee and N. M. Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, 17(11):2094–2107, 2015.
- Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output CNN for age estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.
- Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. Mean-variance loss for deep age estimation from a face. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5285–5294, 2018. doi: 10.1109/CVPR.2018.00554.
- Sebastian Raschka. MLxtend: Providing machine learning and data science utilities and extensions to Python’s scientific computing stack. *Journal of Open Source Software*, 3(24):638, April 2018. doi: 10.21105/joss.00638.
- K. Ricanek and T. Tesafaye. Morph: a longitudinal image database of normal adult age-progression. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 341–345, 2006. doi: 10.1109/FGR.2006.78.
- Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without key-points. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
- C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 397–403, 2013.
- Shizhan Zhu, Cheng Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4998–5006, 2015.
- Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3476–3483, 2013. doi: 10.1109/CVPR.2013.446.
- G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3659–3667, 2015.
- Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Minghui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, PP, April 2020.
- Xinyao Wang, Liefeng Bo, and Fuxin Li. Adaptive wing loss for robust face alignment via heatmap regression. In *2019 IEEE International Conference on Computer Vision (ICCV)*, pp. 6970–6980, 2019. doi: 10.1109/ICCV.2019.00707.
- W. Wu and S. Yang. Leveraging intra and inter-dataset variations for robust face alignment. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2096–2105, 2017.
- W. Wu, Chen Qian, S. Yang, Q. Wang, Y. Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2129–2138, 2018.
- X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 532–539, 2013.
- Zixuan Xu, Banghuai Li, Miao Geng, Ye Yuan, and Gang Yu. Anchorface: An anchor-based facial landmark detector across large poses. *ArXiv:2007.03221*, 2020.

- Tsun-Yi Yang, Yi-Hsuan Huang, Yen-Yu Lin, Pi-Cheng Hsiu, and Yung-Yu Chuang. Ssr-net: A compact soft stagewise regression network for age estimation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI'18*, pp. 1078–1084. AAAI Press, 2018. ISBN 9780999241127.
- Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Computer Vision – ECCV 2014*, pp. 1–16, 2014.
- S. Zhu, C. Li, C. C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3409–3417, 2016. doi: 10.1109/CVPR.2016.371.
- Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 146–155, 06 2016.