Age Prediction Based on Iris Biometric Data

Abstract

In this document, prediction of age for given dataset as classification task where certain ranges of ages represents classes are performed. Different features are are given with the dataset namely texture and geometric features extracted from iris images. We trained different models with different hyperparameters, and analyze the results. The discussion on the dataset, performance and methods are also presented in this report. Best performing models achieved 57.89 % Accuracy on given dataset.

Index Terms:— Iris, Age Prediction, Texture Features, Geometric Features, RELU, Softmax, RMSProp, Categorical CrossEntropy

I. Training with Geometric Features

In the training process with the set of geometric iris features, models are built between zero to four hidden layers with a different set of neurons in each layer. In addition, the geometric feature set contains 5 binary features to given age classes. To detect patterns in higher dimensional spaces, the number of neurons is selected between 8-32. Learning rates are 0.01, 0.005, and 0.001. Also, validation split is done on the training set with the rate of 0.25, i.e., 25% of the training set is sampled to create a validation set. While this sampling is applied, the ratio between each class is preserved to make sure that the validation set represents an equivalent subset of the training set. Activation functions are selected as Rectified Linear Unit (RELU) for the hidden layers and Softmax for the output layer. Categorical Cross Entropy Loss with RMSProp optimizer is also employed as requested in the project requirements. Overall results of all the experiments are provided in Table I.

TABLE I
GEOMETRIC FEATURE RESULTS
HL: HIDDEN LAYER **ACC:** ACCURACY **LR:** LEARNING RATE
EP: EPOCHS **REC:** RECALL, **TA:** TRAINING ACCURACY

HL	LR	N	Ep.	Acc.	Prec.	Rec.	F1	T.A.
0	**0.01**		50	57.9	34	58	42	43.5
			100	57.9	34	58	42	48.7
	0.005		50	57.9	34	58	42	46.4
			100	7	2	3.3	4	45.8
	0.001		50	35.1	12	33	17	23.6
			100	7	0	7	1	45.43
1	**0.01**	**16**	50	57.9	34	58	42	56.5
			100	57.9	34	58	42	58.5
		8	50	57.9	34	58	42	56.5
			100	57.9	34	58	42	56.5
	0.005	**16**	50	57.9	34	58	42	59.8
			100	57.9	34	58	42	59
		8	50	57.9	34	58	42	56.4
			100	57.9	34	58	42	56.1
	0.001	**16**	50	57.9	34	58	42	44.4
			100	57.9	34	58	42	55.2
		8	50	57.9	34	58	42	42
			100	57.9	34	58	42	58.6
2	**0.01**	**32,16**	50	57.9	34	58	42	57.3
			100	57.9	34	58	42	54.3
		16,8	50	57.9	34	58	42	56
			100	57.9	34	58	42	55.1
	0.005	**32,16**	50	57.9	34	58	42	58.7
			100	57.9	34	58	42	56.8
		16,8	50	57.9	34	58	42	60
			100	57.9	34	58	42	56.3
	0.001	**32,16**	50	57.9	34	58	42	55.6
			100	57.9	34	58	42	58.2
		16,8	50	57.9	34	58	42	55.6
			100	57.9	34	58	42	57.6
3	**0.01**	**32,16,8**	50	57.9	34	58	42	55.6
			100	57.9	34	58	42	58.5
		16,8,4	50	57.9	34	58	42	56.7
			100	57.9	34	58	42	57.4
	0.005	**32,16,8**	50	57.9	34	58	42	59
			100	57.9	34	58	42	58.3
		16,8,4	50	57.9	34	58	42	55.9
			100	57.9	34	58	42	54.4
	0.001	**32,16,8**	50	57.9	34	58	42	55.3
			100	57.9	34	58	42	56.2
		16,8,4	50	57.9	34	58	42	56.9
			100	57.9	34	58	42	58.7

A. Results with 4 different Hidden Layer NN

Results for geometric features are shown in Table I.

B. Discussion on Effects of Number of Hidden Layers

In geometric the results of the models vary between 35-58% in terms of accuracy score in test set even though they occasionally perform with nearly 70% score in the training set. The possible reasons for this case are further examined in discussion on the test set. In addition, even though it is not presented in the tables, the accuracy of the model on the validation set reaches up to 72%. On Table I, also written the precision and recall values which presents the quality of the classifier. This is related with accuracy rate. Best performing models for the geometric one are presented in Section V. According to changes on number of hidden layers, epochs or learning rates could not detect a distinctive pattern to discriminate the given classes from each other. The high number of samples belong to second class(ages between 26 and 60), therefore, results indicated that the best performing models on the test set only manages to predict this class.

II. Training with Texture Features

Similar to training process with the previous set of features, models are built between zero to four hidden layers with different set of neurons in each layer. However, as the texture feature set contains 9600 binary features to given age classes, to be able to detect patterns in higher dimensional spaces, number of neurons are selected between 128-1024. Learning rates are the same as geometric features, and validation split is done on training set with the rate 0.2, i.e. 20 % of the training set is sampled to create a validation set. While this sampling is applied, the ratio between each class is preserved to make sure that validation set represents an equivalent subset of the training set. Activation functions are selected as Rectified Linear Unit (RELU) for the hidden layers and Softmax for output layer. Categorical Cross Entropy Loss with RMSProp optimizer is also employed as requested in the project requirements. Overall results of all the experiments are provided in Table II.

A. Results with 4 different Hidden Layer NN

B. Discussion on Effects of Number of Hidden Layers

In contrast to the results with Geometric features presented in Section I, the results of the models vary between 40-58% in terms of

TABLE II
TEXTURE FEATURE RESULTS
HL: HIDDEN LAYER **ACC:** ACCURACY **LR:** LEARNING RATE
EP: EPOCHS **REC:** RECALL, **TA:** TRAINING ACCURACY

HL	LR	# of Neurons	Ep	Acc	Prec	Rec	F1	TA
0	**0.01**		50	53.1	72	58	43	99.9
			100	50.2	46	50	48	100
	0.005		50	54.1	50	55	52	97.7
			100	50.2	47	50	49	100
	0.001		50	50.6	48	51	49	100
			100	53.5	49	54	50	100
1	**0.01**	**1024**	50	57.9	34	58	42	60.3
			100	50.2	45	50	48	73
		512	50	57.9	34	58	42	62
			100	40.3	52	40	32	77
	0.005	**1024**	50	55	49	55	51	92.8
			100	54	49	54	51	94.7
		512	50	49	48	49	48	79.3
			100	53	48	53	50	93
	0.001	**1024**	50	54	48	54	48	99.5
			100	54.4	50	54	52	99.9
		512	50	54.6	50	55	51	94.5
			100	53.3	49	53	51	100
2	**0.01**	**1024,512**	50	57.9	34	58	42	63
			100	57.9	34	58	42	58.9
		512,256	50	57.9	34	48	42	54.9
			100	57.9	34	58	42	57.49
	0.005	**1024,512**	50	52.8	41	53	44	69.7
			100	57.9	34	58	42	59.3
		512,256	50	57.9	34	58	42	59.7
			100	54.4	49	54	51	83.6
	0.001	**1024,512**	50	56.1	48	56	49	95.1
			100	51.7	49	52	50	99.1
		512,256	50	49.3	46	49	48	99.2
			100	55.9	50	56	53	99.9
3	**0.01**	**1024,512,256**	50	57.9	47	57	44	58.8
			100	57	47	57	44	63.7
		512,256,128	50	57.9	34	58	42	57.9
			100	57.9	34	58	42	57.7
	0.005	**1024,512,256**	50	44	46	44	43	67.2
			100	38.6	43	39	35	96.3
		512,256,128	50	57.4	42	57	43	74.3
			100	42.5	46	43	39	79.2
	0.001	**1024,512,256**	50	57.9	34	58	42	98
			100	51.8	48	52	50	99.7
		512,256,128	50	51	47	51	49	87.1
			100	54.8	50	55	52	97.8

accuracy score in test set even though they occasionally perform with nearly 100% score in training set. The possible reasons for this case is further examined in discussion on test set in Section V. In addition, even though it is not presented in the tables, the accuracy of the model on validation set reaches up to 86% which has no beneficial effect of the performance of the model on test set. Table II also represents the weighted average values of precision and recall values which presents the quality of the classifier. However, as the given result is weighted, it favors the classes based on the samples they have. Further discussion on the overall precision and recall values of the best performing models are presented in Section V All in all, despite the high number of features presented in texture features, the developed models with various number of hidden layers could not detect distinctive pattern to discriminate the given classes from each other. Further examination on the results indicated that the best performing models on test set only manages to predict the second class (ages between 26 and 60) due to the high number of samples belong to that class.

III. TRAINING WITH GEOMETRIC AND TEXTURE FEATURES

A. *Results with 4 different Hidden Layer NN*

Models are built between zero to four hidden layers with different set of neurons in each layer. As the combination of texture and geometric feature set contains 9605 binary features to be able to detect patterns in higher dimensional spaces, number of neurons are selected between128-1024. Learning rates are the same as the previous two features, and validation split is done on training set. 20 %of the training set is sampled to create a validation set. While this sampling is applied, the ratio between each class is preserved to make sure that validation set represents an equivalent subset of the training set. Activation functions are selected as Rectified Linear Unit (RELU)for the hidden layers and Softmax for output layer. Categorical CrossEntropy Loss with RMSProp optimizer is also employed as requested in the project requirements.

TABLE III
COMBINATION OF GEOMETRIC AND TEXTURE FEATURE RESULTS
HL: HIDDEN LAYER **ACC:** ACCURACY **LR:** LEARNING RATE
EP: EPOCHS **REC:** RECALL, **TA:** TRAINING ACCURACY

HL	LR	N	Ep.	Acc.	Prec.	Rec.	F1	T.A.
0	**0.01**		50	57.9	34	58	49.4	49.4
			100	43.2	49	43	40	55.8
	0.005		50	55.7	50	56	52	44.7
			100	09	53	09	06	63.8
	0.001		50	17.9	50	18	13	54.6
			100	35.1	12	35	18	79.5
1	**0.01**	**1024**	50	57.9	34	58	42	58.2
			100	57.9	34	58	42	55
		512	50	57.9	34	58	42	57.7
			100	57.9	34	58	42	57.8
	0.005	**1024**	50	57.9	34	58	42	59.7
			100	57.9	34	58	42	57.2
		512	50	57.9	34	58	42	57.5
			100	57.9	34	58	42	60
	0.001	**1024**	50	57.9	34	58	42	57.3
			100	57.9	34	58	42	57.3
		512	50	57.9	34	58	42	57.4
			100	47.6	43	48	45	61.5
2	**0.01**	**1024,512**	50	57.9	34	58	42	57.3
			100	57.9	34	58	42	57.3
		512,256	50	57.9	34	58	42	57.3
			100	57.9	34	58	42	57.3
	0.005	**32,16**	50	57.9	34	58	42	57.3
			100	57.9	34	58	42	57.3
		16,8	50	57.9	34	58	42	57.3
			100	57.9	34	58	42	57.3
	0.001	**32,16**	50	57.9	34	58	42	56.5
			100	57.9	34	58	42	57.6
		16,8	50	57.9	34	58	42	57.1
			100	57.9	34	58	42	57.5
3	**0.01**	**32,16,8**	50	57.9	34	58	42	57.4
			100	57.9	34	58	42	57.4
		16,8,4	50	57.9	34	58	42	55.5
			100	57.9	34	58	42	56.5
	0.005	**32,16,8**	50	57.9	34	58	42	55.9
			100	57.9	34	58	42	55.6
		16,8,4	50	57.9	34	58	42	58
			100	57.9	34	58	42	57.4
	0.001	**1024,512,256**	50	57.9	34	58	42	57.4
			100	57.9	34	58	42	57.4
		512,256,128	50	57.9	34	58	42	57.4
			100	57.9	34	58	42	57.4

B. *Discussion on Effects of Number of Hidden Layers*

In this section, we have examined the concatenated features of Geometric and Texture Features. The accuracy values we get in this model vary between 35% and 58%, and in general, 58% accuracy is obtained. The model we created generally values between 50% and 60% and is consistent with the actual results we found. The results we found in this section are in agreement with the actual values we received, unlike the Texture Feature set. F1 recall and precision values are also given in the table above. The results here are given as weighted. We tried the model we created in different configurations with various layer numbers and different neuron numbers in each layer. These configurations were trained with different epoch values using 0-4 hidden layers and using 128-1024 nodes in each hidden layer. The training set we used was used as a Validation set by 20%. Each result from this model is shown in the table above.Among these models, we get the lowest value with 9% results with 0.005 Learning rate and 100 epochs, while we get the best value with 1 Hidden layer and 512 nodes for 57.9%. Apart from these, we generally get a value of 57.9% regardless of Hidden Layer. Looking at these results, we can say that our model tends to prefer the 2nd class because we have a large number of class2 samples. This issue will be addressed again in Section V.

IV. Results

Best Configurations and Results on Test Set

TABLE IV
BEST CONFIGURATION RESULTS
HL: HIDDEN LAYER **ACC:** ACCURACY
REC: RECALL, **TA:** TRAINING ACCURACY

Type	HL	Acc	Prec	Rec	F1	TA
Geometric	0	57.9	34	58	42	48.7
	1	57.9	34	58	42	59.8
	2	57.9	34	58	42	60
	3	57.9	34	58	42	59
Texture	0	54.1	50	55	52	97.7
	1	57.9	34	58	42	60.3
	2	57.9	34	58	42	63
	3	57.9	34	58	42	98
Combination	0	57.9	34	58	49.4	49.4
	1	57.9	34	58	42	60
	2	57.9	34	58	42	57.6
	3	57.9	34	58	42	58

V. Additional Comments and Discussion

As presented throughout all of the sections, the models couldn't perform above a certain accuracy score. To understand this behaviour the studies [**?**] and [1] are examined as well as the proposed dataset. In the explanation of the dataset, it is clearly presented that the number of each age group does not contain equal number of participants. This creates an imbalanced dataset problem for our model development. Additionally, even though it is not limited, the total number of samples in the given dataset is 1144 which contains multiple samples from same subjects. These two cases are the initial possible problems that requires additional testing. For the former, one of the popular methods to tackle was class weighting. It means giving a weight to the activations of a class with low samples to mitigate or overcome to effect of imbalance. For this purpose, we try the weights $\left[\frac{115}{70}, \frac{115}{115}, \frac{115}{15}\right]$ for class with participants $[70, 115, 15]$ some of the best models we obtained. However, it did not overcome the problem. Subsequently, we have tried balancing the dataset using resampling, where the samples from the classes with lower sample sizes are randomly selected with replacement, i.e. duplicated. Even though this seemed it can solve the problem, due to the huge gap between the maximum and minimum number of samples among classes, the process end up with replicating the entire age group more than once to balance the classes, which confuse the best performing models.

For the latter problem, as the extracted features are provided with limited details which are not matching with the background that the team posses, we have examined the results of [**?**], [1]. In [**?**], authors employed the same approaches which we provide in this document. For geometric features, they indicate that they use the five uncorrelated geometric features which matches with the feature vector that we have. Similarly, for texture, paper indicates they have 6 features with 780 value in each one for each iris. This ends up with 4680 value for each eye, i.e. a vector with 9360 values when combined. Even though we have 9600 features in the provided dataset, the dimensionality of the given data is similar. Considering the models used by the authors, MLP, which is what is used in this study, achieves 57.86%, 67.86%, 76.64% accuracy for only geometric, only texture and geometric and texture combined features respectively. However, as the study does not indicate the architectural detail of the proposed MLP, we were unable to detect how the mentioned problems are solved. Additionally, as only accuracy metric is presented, we were unable to see the classifiers' performance on each class individually.

To see the performance of our classifiers in detail, we have examined each classes accuracy (recall) individually, and observed that our models with higher accuracy focuses on learning single class with highest number of samples. This case shows that imbalance in dataset forces model to optimize its performance based on single class. Surprisingly, the models learn to classify more than one class performed worse than the best performing model. From that observation, we commented that the features are extracted might not be sufficient to distinguish between given classes with MLP. To support this idea, [**?**], shows that when multiple models are used, performance increases significantly.

Apart from the given methods and discussions, we have tested models with different batch sizes between 16-256, with extremely low learning rates such as 5×10^{-5} and with L1, L2 Regularization. Also, feature elimination based on the correlation with the target variable is tested. Unfortunately, the models tested performs up to 58% Accuracy.

All in all in this project, we had a chance to develop a multi class classifier and examine its performance under different settings with an imbalanced dataset. Even though the obtained performance for the classifiers are maximum 58%, the overall experience was quite beneficial in terms of exploring a real life dataset, and developing machine learning models with Tensorflow Framework[2] and Keras API[3].

References

[1] M. Erbilek, M. Fairhurst, and M. C. D. C. Abreu. Age prediction from iris biometrics. In *5th International Conference on Imaging for Crime Detection and Prevention (ICDP 2013)*, pages 1–5, 2013.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, and howpublished = "https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac" note = "Accessed: 2021-6-20" = feb, year = 2019. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[3] Keras Team. Keras.io".