HIGH SCHOOL EDITION
Journal of Student Research

Volume 12 Issue 3 (2023)

Quantifying the Dynamics of Data Augmentation Hyperparameters for Facial Expression Recognition

Ryan Lin[1] and Peter Washington[#]

[1]Polygence
[#]Advisor

ABSTRACT

Automated recognition of facial expressions is a central component of systems used in an expanding array of domains. For a computer to automatically recognize affect, copious amounts of data are required to successfully train the model. It can often take a lot of work to collect and label data. In recent years, researchers have applied numerous data augmentation strategies to increase the diversity of the data within training datasets. Here, I examined the most common data augmentation strategies to determine which strategies result in higher performance for the facial expression recognition machine learning model. I first tested each data augmentation technique by itself and compared their performances. I next ran an ablation study with the augmentation strategies. I then analyzed the effect of dataset size on the marginal contribution of data augmentation. I find that augmentation does not always improve performance. When the dataset size is small, it results in a degradation of model performance. The accuracy of models with data augmentation starts to outperform the models with no data augmentation when the training dataset size is greater than a certain threshold. These results highlight the importance of considering dataset size when applying data augmentation to computer vision.

Introduction

Facial Expression Recognition (FER) using machine learning is widely researched and applied to many areas, such as human-robot interaction [16-20], digital therapeutics for behavioral healthcare conditions such as autism [21-29], diagnostics for behavioral healthcare [30-31], driver safety [32-38], and many other domains. To robustly train an automated model for facial expression recognition, I need variety within the dataset. Collecting and labeling such data can be time-consuming and costly. Data collection could also bring up privacy concerns, which can make it hard for it to reach a massive scale.

Data augmentation is a technique that encompasses the artificial expansion of a base dataset by using data preprocessing techniques, thereby introducing variability into the dataset [39-40]. Data augmentation can therefore improve the performance of FER machine learning models. Data augmentation has been frequently applied to FER, usually bolstering the performance when compared to no data augmentation [1] [2] [3] [5] [10-15].

While data augmentation is a nearly universal step in the machine learning computer vision model training pipeline, a formal study of the effects of data augmentation at various dataset sizes has not been published. It is important to understand when data augmentation may help a model generalize versus when the augmentation may actually hurt (i.e., when the dataset is prohibitively small).

To address this gap in the FER literature, I compare commonly applied data augmentation strategies for computer vision processing. I compare the performance of FER models when training with various training set sizes both with and without data augmentation. I then analyze the effect of data augmentation and training dataset size on the FER model accuracy, precision, and recall. I find that with small dataset sizes, data augmentation hurts the training of FER models, but this effect is reversed when the training set size is above roughly 1750 images. This work suggests that data augmentation is most useful with medium to large datasets and can actually hurt when the dataset is small. I conjecture that this is because at smaller dataset sizes, data augmentation “confuses” the model due to the difference in data distribution between the train and test sets.

Related Work

Data augmentation is used in Facial Expression Recognition (FER) to introduce more variations into the datasets for training the machine learning models to be more effective and accurate. Several types of data augmentation strategies are developed or chosen by researchers to improve the performance of their Facial Expression Recognition models. Data augmentation is also used to avoid overfitting the models due to insufficient training datasets.

Geometric transformations are one of the most common data augmentation strategies chosen. It includes rotation, reflection, flips, shifts, shears, and scale [5]. Some oversampling augmentations are also developed or used, such as Generative Adversarial Network (GAN) [2] [9]. Image pixel editing data augmentation strategies like random cropping, erasing, random erasing [3], random noise [8], skew, and occlusion are considered to enhance CNN models for FER [4] [5] [8]. Some chose the image attribute changing data augmentation strategies such as adjusting illumination [6], contrast [4] [6] [8], or adding color jittering to create additional training images. Most researchers chose more than 1 data augmentation strategy for creating more varieties and avoiding overfitting [1] [2] [4] [5] [6] [7]. Various techniques are derived from some data augmentation strategies. For example, Random Erasing can be done image-aware, object-aware, or image and object-aware [3]. A Point Adversarial Self Mining (PASM) approach was Inspired by random erasing and adversarial erasing [7]. In [8], Face Detection was combined with data augmentation techniques to achieve better performance for Facial Expression Recognition.

Almost all FER models achieve better accuracy after training with augmented datasets. Most improvements are significant. A few of them see significant improvement on smaller datasets, but no significant improvement on already large datasets, e.g. CelebA [4]. Models trained with augmented datasets may also achieve higher accuracy with fewer epochs [5].

Methods

Dataset Description

I tested my method on a widely used facial expression recognition dataset, the Facial Expression Recognition 2013 dataset (FER2013). FER2013 data are collected in the wild. Images in the dataset contain variations in pose, view angle, and lighting condition. There are 35,887 48x48 grayscale images with 7 emotions labeled as Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The 3 majority classes are Happy: 8,989 images, Neutral: 6,198 images, and Sad: 6,077 images. I only selected these three classes, Happy, Neutral, and Sad, for my experiment. The total number of images from these three classes is 21,264. During the experiment, 90% of the images were selected as training data, the remaining 10% of the images were used for model validation. So the training dataset has 19,137 images, and the validation dataset has 2,127 images.

Table 1. Emotional label distribution in the FER2013 dataset.

Emotion	Number of images
Happy	8,989
Neutral	6,198

Sad	6,077
Fear	5,121
Angry	4,953
Surprise	4,002
Disgust	547

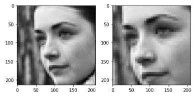
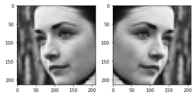
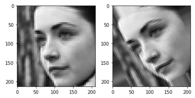
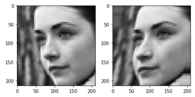
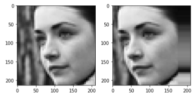
Machine Learning Model Details

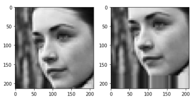
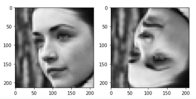
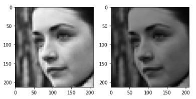
I implemented a 2D convolutional neural network (CNN) with 6 convolutional layers, each with max pooling and ELU activation. I applied dropout to the final dense layers of the network for regularization. Because the point of this study is to compare data augmentation strategies rather than to develop the highest performing model, I implemented a single CNN without hyperparameter optimization.

Data Augmentation Strategies

The training image data is augmented by ImageDataGenerator in the Keras API. I used 8 common augmentation strategies provided in the ImageDataGenerator class: zoom, horizontal flip, rotation, shear, horizontal shift, vertical shift, vertical flip, and brightness level. These strategies are described further and examples are shown in Table 2.

Table 2. Data augmentation strategies and examples of modified data points using each strategy.

Data Augmentation Strategy	Sample Images
Zoom: Generate images with varying zoom-in or zoom-out levels.	
Horizontal flip: Flip or mirror an image in the horizontal direction (left-right)	
Rotation: Rotate an image a certain degree	
Shear: Slant an image. Shearing is also known as skewing	
Width shift: Horizontal shift of the pixels of the image without changing the dimension of the image.	

Height shift: Vertically shift the pixels of the image without changing the dimension of the image.	
Vertical flip: Flip or mirror an image in the vertical direction (up-down)	
Brightness: Generate images with varied brightness levels.	

I explored a variety of data augmentation hyperparameters per strategy using grid search. All data augmentation strategies and corresponding hyperparameters explored are listed in Table 3.

Table 3. Data augmentation strategies and hyperparameters.

Data Augmentation	Parameter values

Zoom	zoom_range=0.15
	zoom_range=0.25
	zoom_range=0.35
	zoom_range=0.50
	zoom_range=0.75

Horizontal flip	True, False

Rotation	rotation_range=15
	rotation_range=30
	rotation_range=45
	rotation_range=60
	rotation_range=75

Shear	shear_range=0.15
	shear_range=0.30
	shear_range=15.0
	shear_range=30.0
	shear_range=45.0

Horizontal shift	width_shift_range=0.15
	width_shift_range=0.30
	width_shift_range-0.45

Vertical shift	height_shift_range=0.15
	height_shift_range=0.30
	height_shift_range=0.45

Vertical flip	True, False

Brightness level	brightness_range(0.3, 0.9)

Experimental Procedures

The first set of experiments began with training my FER model without any data augmentation and recording the validation result as a baseline to compare against. To identify which data augmentation strategy is most effective, I ran a series of experiments by using only one data augmentation strategy out of the eight common data augmentation strategies I included in the study (listed in Table 2) to train my FER model and compare the result with the baseline ("only-include-one study"). The validation accuracy of adjusting the brightness level was much lower than other data augmentation strategies. I therefore excluded the brightness range hyperparameter in the following experiments.

To compare how much the performance changes when I exclude one of the data augmentation strategies, I trained the FER model with seven out of eight of the common data augmentation strategies applied and removed a single strategy ("ablation study").

In the third set of experiments, I used hyperparameter optimization to identify which parameter for each augmentation strategy yielded higher accuracy. I used one data augmentation strategy at a time, replaced the parameters with different values, and recorded the results. For example, multiple experiments were conducted for rotating images with 15, 30, 45, 60, and 75 degrees to compare the performance.

For the final set of experiments, I trained the FER model with various training dataset sizes without any data augmentation, and with seven common data augmentation strategies applied. I then generated 100 sets of 1,500 randomly selected validation samples and sent them to the trained FER models for prediction. I then compared the performance to analyze the effect of data augmentation for different training dataset sizes. The experimental procedures are illustrated in Figure 1.

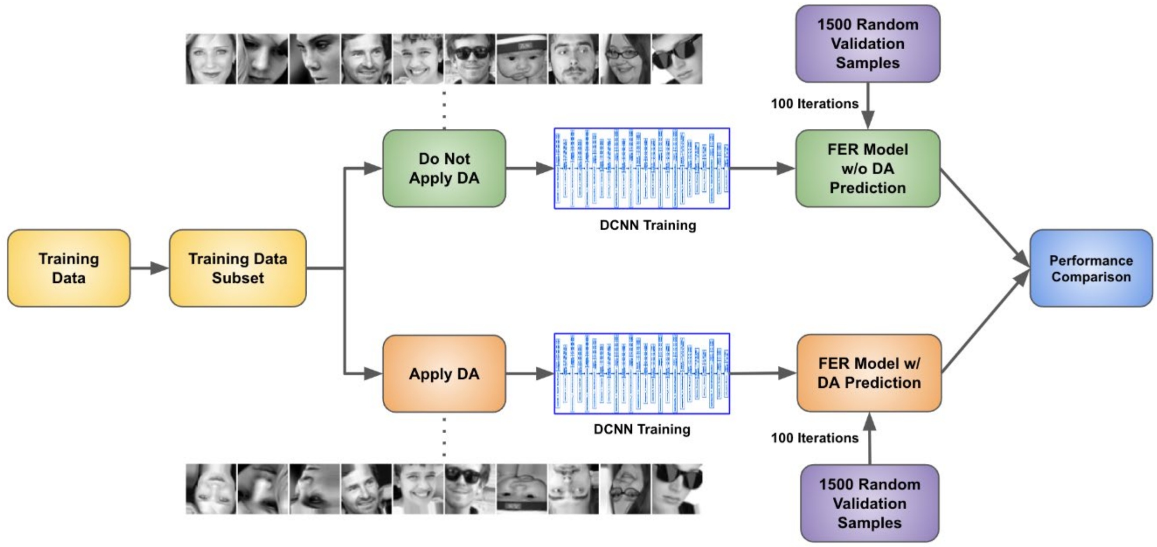


Figure 1. Illustration of the experimental procedures. I compare the performance of a model with and without data augmentation on the FER2013 dataset.

Results

The mean validation accuracy of the FER model trained without any data augmentation is 0.8132. The results of the only-include-one analysis (Table 2 and Table 3) indicate that each DA strategy produced about the same range of accuracy, precision, and recall (the mean accuracy is 0.8094, the mean precision is 0.8158, and the mean recall is

0.8009) except the brightness level. Brightness augmentation resulted in much lower accuracy (0.4423), so I excluded brightness level in the following experiments. The brightness level was an outlier. Vertical shift and horizontal flip have the best accuracy (0.8197 and 0.8180 respectively) whereas vertical flip and shear have the lowest accuracy (0.7965 and 0.7993 respectively.) The results are listed in Table 4.

Table 4. FER model validation results for only-include-one experiments.

DA	Performance
None	accuracy mean = **0.813253**, stdev = 0.004791, range = [0.808462, 0.818044] precision mean = 0.813845, stdev = 0.004785, range = [0.809060, 0.818630] recall mean = 0.810487, stdev = 0.004865, range = [0.805622, 0.815352]
Zoom (range=0.15)	accuracy mean = 0.816100, stdev = 0.004980, range = [0.811120, 0.821080] precision mean = 0.823098, stdev = 0.004948, range = [0.818150, 0.828046] recall mean = 0.811920, stdev = 0.005025, range = [0.806895, 0.816945]
Horizontal flip	accuracy mean = 0.818060, stdev = 0.004501, range = [0.813559, 0.822561] precision mean = 0.824090, stdev = 0.004387, range = [0.819703, 0.828477] recall mean = 0.811020, stdev = 0.004490, range = [0.806530, 0.815510]
Rotation (range=15)	accuracy mean = 0.802820, stdev = 0.006090, range = [0.796730, 0.808910] precision mean = 0.823529, stdev = 0.005991, range = [0.817538, 0.829520] recall mean = 0.772247, stdev = 0.006478, range = [0.765769, 0.778724]
Shear (range=0.15)	accuracy mean = 0.799313, stdev = 0.005082, range = [0.794231, 0.804395] precision mean = 0.806001, stdev = 0.005036, range = [0.800965, 0.811037] recall mean = 0.787000, stdev = 0.005005, range = [0.781995, 0.792005]
Horizontal shift (range=0.15)	accuracy mean = 0.809613, stdev = 0.005418, range = [0.804196, 0.815031] precision mean = 0.814779, stdev = 0.005373, range = [0.809406, 0.820153] recall mean = 0.806300, stdev = 0.005420, range = [0.800880, 0.811720]
Vertical shift (range=0.15)	accuracy mean = **0.819747**, stdev = 0.004839, range = [0.814908, 0.824585] precision mean = 0.821994, stdev = 0.004881, range = [0.817113, 0.826874] recall mean = 0.815007, stdev = 0.004948, range = [0.810058, 0.819955]
Vertical flip	accuracy mean = **0.796507**, stdev = 0.005580, range = [0.790926, 0.802087] precision mean = 0.798847, stdev = 0.005604, range = [0.793244, 0.804451] recall mean = 0.793160, stdev = 0.005683, range = [0.787477, 0.798843]
Brightness (range: 0.3 - 0.9)	accuracy mean = **0.442340**, stdev = 0.006494, range = [0.435846, 0.448834] precision mean = 0.476429, stdev = 0.009094, range = [0.467336, 0.485523] recall mean = 0.282227, stdev = 0.006353, range = [0.275874, 0.288579]

The mean validation accuracy of my FER model trained with all data augmentation is 0.8130, the mean precision is 0.8259, and the mean recall is 0.7945. The mean accuracy of the ablation study is 0.8191, the mean precision is 0.8309, and the mean recall is 0.8043. Removing vertical flip resulted in the optimal mean validation accuracy of 0.8275. Removing rotation resulted in the lowest mean validation accuracy of 0.8147. The results are listed in Table 5.

Table 5. FER model validation results for ablation study.

DA	Performance
All 7 DA	accuracy mean = **0.813007**, stdev = 0.005316, range = [0.807690, 0.818323] precision mean = 0.825921, stdev = 0.005068, range = [0.820853, 0.830989] recall mean = 0.794453, stdev = 0.005411, range = [0.789042, 0.799864]
No zoom	accuracy mean = 0.816433, stdev = 0.005758, range = [0.810675, 0.822192] precision mean = 0.834433, stdev = 0.005519, range = [0.828914, 0.839952] recall mean = 0.800333, stdev = 0.005936, range = [0.794397, 0.806270]
No horizontal flip	accuracy mean = 0.816447, stdev = 0.006759, range = [0.809688, 0.823205] precision mean = 0.827865, stdev = 0.006244, range = [0.821621, 0.834110] recall mean = 0.804607, stdev = 0.006895, range = [0.797711, 0.811502]
No rotation	accuracy mean = **0.814720**, stdev = 0.005152, range = [0.809568, 0.819872] precision mean = 0.832193, stdev = 0.005167, range = [0.827026, 0.837360] recall mean = 0.796493, stdev = 0.005707, range = [0.790787, 0.802200]
No shear	accuracy mean = 0.820160, stdev = 0.005494, range = [0.814666, 0.825654] precision mean = 0.829786, stdev = 0.005230, range = [0.824556, 0.835016] recall mean = 0.804660, stdev = 0.005537, range = [0.799123, 0.810197]
No horizontal shift	accuracy mean = 0.822153, stdev = 0.005760, range = [0.816394, 0.827913] precision mean = 0.831563, stdev = 0.005705, range = [0.825858, 0.837269] recall mean = 0.810553, stdev = 0.005946, range = [0.804608, 0.816499]
No vertical shift	accuracy mean = 0.816680, stdev = 0.004962, range = [0.811718, 0.821642] precision mean = 0.826740, stdev = 0.004901, range = [0.821839, 0.831641] recall mean = 0.803607, stdev = 0.004989, range = [0.798618, 0.808596]
No vertical flip	accuracy mean = **0.827567**, stdev = 0.004721, range = [0.822846, 0.832288] precision mean = 0.834333, stdev = 0.004829, range = [0.829504, 0.839163] recall mean = 0.810500, stdev = 0.005136, range = [0.805364, 0.815636]

The results of hyperparameter optimization for one data augmentation at a time are listed in Table 6. The mean validation accuracy is 0.8120. The mean precision is 0.8198. The mean recall is 0.8019. The best validation accuracy is 0.8312 with a rotation range of 30. The lowest validation accuracy is 0.7993 with a shear range of 0.15.

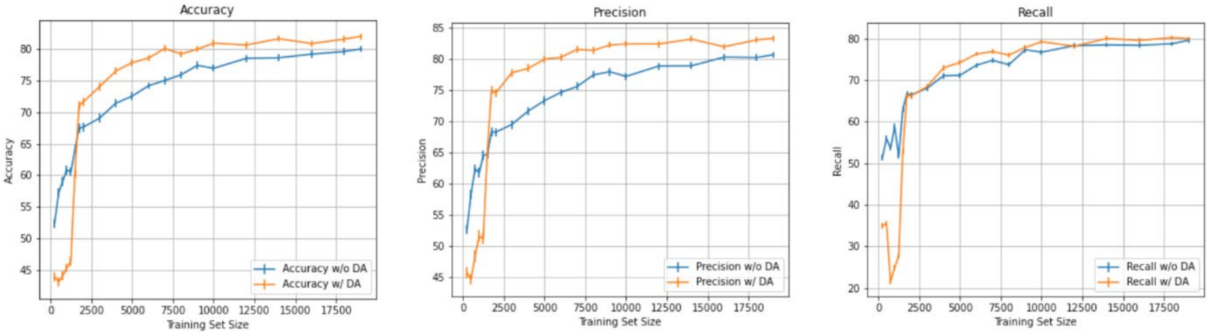
Table 6. FER model validation results for hyperparameter optimization for only-include-one experiments.

DA	Parameter values	Performance
Zoom	0.15	accuracy mean = 0.809513, stdev = 0.005208, range = [0.804306, 0.814721] precision mean = 0.818138, stdev = 0.005332, range = [0.812807, 0.823470] recall mean = 0.801633, stdev = 0.005375, range = [0.796258, 0.807008]

	0.25	accuracy mean = 0.812673, stdev = 0.004963, range = [0.807710, 0.817637] precision mean = 0.821708, stdev = 0.004831, range = [0.816876, 0.826539] recall mean = 0.797653, stdev = 0.004972, range = [0.792681, 0.802626]
	0.35	accuracy mean = 0.812073, stdev = 0.005240, range = [0.806833, 0.817313] precision mean = 0.827452, stdev = 0.005226, range = [0.822226, 0.832679] recall mean = 0.800653, stdev = 0.005367, range = [0.795287, 0.806020]
	0.50	accuracy mean = 0.821340, stdev = 0.005570, range = [0.815770, 0.826910] precision mean = 0.827328, stdev = 0.005457, range = [0.821872, 0.832785] recall mean = 0.810893, stdev = 0.005771, range = [0.805123, 0.816664]
	0.75	accuracy mean = 0.818193, stdev = 0.005226, range = [0.812968, 0.823419] precision mean = 0.825749, stdev = 0.005109, range = [0.820640, 0.830857] recall mean = 0.804940, stdev = 0.005417, range = [0.799523, 0.810357]
Rotation	15	accuracy mean = 0.802820, stdev = 0.006090, range = [0.796730, 0.808910] precision mean = 0.823529, stdev = 0.005991, range = [0.817538, 0.829520] recall mean = 0.772247, stdev = 0.006478, range = [0.765769, 0.778724]
	30	accuracy mean = **0.831253**, stdev = 0.005277, range = [0.825977, 0.836530] precision mean = 0.833038, stdev = 0.005348, range = [0.827690, 0.838386] recall mean = 0.825673, stdev = 0.005268, range = [0.820405, 0.830942]
	45	accuracy mean = 0.812207, stdev = 0.005793, range = [0.806413, 0.818000] precision mean = 0.827020, stdev = 0.005625, range = [0.821394, 0.832645] recall mean = 0.801100, stdev = 0.005771, range = [0.795329, 0.806871]
	60	accuracy mean = 0.806360, stdev = 0.004890, range = [0.801470, 0.811250] precision mean = 0.814915, stdev = 0.005049, range = [0.809866, 0.819964] recall mean = 0.796513, stdev = 0.005074, range = [0.791440, 0.801587]
	75	accuracy mean = 0.811273, stdev = 0.005445, range = [0.805829, 0.816718] precision mean = 0.820934, stdev = 0.005352, range = [0.815582, 0.826285] recall mean = 0.800393, stdev = 0.005441, range = [0.794953, 0.805834]
Shear	0.15	accuracy mean = **0.799313**, stdev = 0.005082, range = [0.794231, 0.804395] precision mean = 0.806001, stdev = 0.005036, range = [0.800965, 0.811037] recall mean = 0.787000, stdev = 0.005005, range = [0.781995, 0.792005]
	0.30	accuracy mean = 0.804453, stdev = 0.005612, range = [0.798842, 0.810065] precision mean = 0.810242, stdev = 0.005469, range = [0.804773, 0.815712] recall mean = 0.800233, stdev = 0.005561, range = [0.794672, 0.805795]
	15.0	accuracy mean = 0.811673, stdev = 0.005270, range = [0.806403, 0.816943] precision mean = 0.814043, stdev = 0.005189, range = [0.808853, 0.819232] recall mean = 0.807840, stdev = 0.005293, range = [0.802547, 0.813133]
	30.0	accuracy mean = 0.810660, stdev = 0.005703, range = [0.804957, 0.816363] precision mean = 0.817170, stdev = 0.005551, range = [0.811618, 0.822721]

		recall mean = 0.799840, stdev = 0.005627, range = [0.794213, 0.805467]
	45.0	accuracy mean = 0.813853, stdev = 0.005276, range = [0.808577, 0.819129] precision mean = 0.817617, stdev = 0.005373, range = [0.812244, 0.822990] recall mean = 0.806547, stdev = 0.005509, range = [0.801038, 0.812056]
Horizontal shift	0.15	accuracy mean = 0.805760, stdev = 0.005170, range = [0.800590, 0.810930] precision mean = 0.814972, stdev = 0.005095, range = [0.809876, 0.820067] recall mean = 0.791507, stdev = 0.005341, range = [0.786165, 0.796848]
	0.30	accuracy mean = 0.809767, stdev = 0.005462, range = [0.804305, 0.815229] precision mean = 0.810901, stdev = 0.005471, range = [0.805430, 0.816372] recall mean = 0.805447, stdev = 0.005570, range = [0.799877, 0.811016]
	0.45	accuracy mean = 0.804540, stdev = 0.005102, range = [0.799438, 0.809642] precision mean = 0.816413, stdev = 0.005178, range = [0.811235, 0.821591] recall mean = 0.793693, stdev = 0.005049, range = [0.788645, 0.798742]
Vertical shift	0.15	accuracy mean = 0.819747, stdev = 0.004839, range = [0.814908, 0.824585] precision mean = 0.821994, stdev = 0.004881, range = [0.817113, 0.826874] recall mean = 0.815007, stdev = 0.004948, range = [0.810058, 0.819955]
	0.30	accuracy mean = 0.820880, stdev = 0.005249, range = [0.815631, 0.826129] precision mean = 0.826343, stdev = 0.005188, range = [0.821154, 0.831531] recall mean = 0.812213, stdev = 0.005483, range = [0.806731, 0.817696]
	0.45	accuracy mean = 0.815460, stdev = 0.005269, range = [0.810191, 0.820729] precision mean = 0.820640, stdev = 0.005355, range = [0.815285, 0.825995] recall mean = 0.809627, stdev = 0.005350, range = [0.804276, 0.814977]

The validation accuracy, precision, and recall attained by the FER model using various training data set sizes with no data augmentation, and with seven common data augmentation strategies applied, are shown in Figure 2. The comparison of accuracy, precision, and recall are shown in Table 7.1, Table 7.2, and Table 7.3. I started with a training dataset with 250 images. The mean validation accuracy is 0.5233 without data augmentation. When I applied all data augmentation strategies, the mean validation accuracy decreased to 0.4394, which is much lower than the accuracy when no data augmentation was applied. When the training dataset size increases, the validation accuracy also increases for FER models with no data augmentation. However, the validation accuracy for the FER models with data augmentation remain within the same range [0.4394 - 0.4638] until the training dataset size is around 1,250, while the validation accuracy of the FER model with no data augmentation increases to 0.605. The accuracy of the FER model with data augmentation increases to 0.6035 when the dataset size is 1,500. The FER model with data augmentation starts performing better than the FER model without data augmentation when the dataset size is above 1,500, around 1,750. The model continues to obtain better performance when the dataset size increases. The performance difference in Table 7.1 demonstrates the biggest improvement (around 7%), which occurs when the dataset size is between 3,000 to 7,000 images. Although I continue to observe better performance while the dataset size increases, the improvement gradually decreases with increasing dataset sizes.



(2a) (2b) (2c)

Figure 2. (a) Model accuracy and corresponding standard deviation. (b) Model precision and corresponding standard deviation. (c) Model recall and corresponding standard deviation.

The precision difference listed in Table 7.2 shows a similar pattern. The precision when I use all seven data augmentation methods with a dataset of 1,500 images or less is worse than the precision obtained without using data augmentation. When the training set size is 1,750 images, the precision when the data augmentation methods are used starts to increase compared to when not applying the corresponding strategies. The precision improved slightly more than accuracy; the highest improvement in the precision was 11.94% while 7.34% for accuracy.

The recall attained by the FER model with data augmentation applied exceeded the recall without data augmentation when the training data set size was 3,000. However, the improvement was not as significant as the improvement with accuracy and precision. The highest percentage of improvement is 4.32%.

Table 7.1. FER model validation accuracy for various training set sizes with and without DA applied.

Training set size	Accuracy w/ DA	Accuracy w/o DA	Difference	Improvement %
250	43.94	52.33	-8.39	-16.03%
500	43.11	57.20	-14.09	-24.63%
750	44.01	59.06	-15.05	-25.49%
1000	45.32	60.86	-15.54	-25.54%
1250	46.38	60.51	-14.13	-23.35%
1500	60.35	64.11	-3.76	-5.87%
1750	**71.10**	**67.45**	**3.65**	**5.41%**
2000	71.61	67.64	3.96	5.86%
3000	74.04	69.05	**4.99**	**7.23%**
4000	76.53	71.43	**5.10**	**7.14%**
5000	77.88	72.55	**5.33**	**7.34%**
6000	78.59	74.20	**4.39**	**5.91%**

7000	80.13	75.05	**5.08**	**6.77%**
8000	79.26	75.96	3.31	4.35%
9000	80.02	77.44	2.58	3.33%
10000	80.96	76.99	3.98	5.17%
12000	80.68	78.56	2.11	2.69%
14000	81.67	78.65	3.02	3.84%
16000	80.90	79.23	1.68	2.12%
18000	81.61	79.64	1.97	2.48%
19000	82.04	80.04	2.00	2.50%

Table 7.2. FER model validation precision for various training set sizes with and without DA applied.

Training set size	**Precision w/ DA**	**Precision w/o DA**	**Difference**	**Improvement %**
250	45.72	52.63	-6.91	-13.13%
500	44.65	58.34	-13.68	-23.46%
750	48.35	62.37	-14.02	-22.48%
1000	51.59	61.71	-10.13	-16.41%
1250	51.14	64.51	-13.37	-20.72%
1500	**65.21**	**64.62**	**0.59**	**0.91%**
1750	74.95	68.32	**6.63**	**9.70%**
2000	74.47	68.21	**6.26**	**9.17%**
3000	77.77	69.47	**8.30**	**11.94%**
4000	78.44	71.62	**6.82**	**9.53%**
5000	79.94	73.29	**6.65**	**9.07%**
6000	80.23	74.60	5.63	7.54%
7000	81.50	75.55	5.94	7.87%
8000	81.34	77.42	3.92	5.06%
9000	82.21	77.92	4.29	5.50%
10000	82.37	77.17	5.20	6.74%

12000	82.37	78.83	3.54	4.49%
14000	83.15	78.87	4.28	5.43%
16000	81.92	80.26	1.66	2.07%
18000	83.03	80.19	2.83	3.53%
19000	83.26	80.65	2.62	3.24%

Table 7.3. FER model validation recall for various training set sizes with and without DA applied.

Training set size	**Recall w/ DA**	**Recall w/o DA**	**Difference**	**Improvement %**
250	34.92	51.38	-16.46	-32.04%
500	35.56	55.92	-20.37	-36.42%
750	21.51	53.65	-32.14	-59.91%
1000	24.93	58.72	-33.80	-57.55%
1250	27.68	51.75	-24.07	-46.51%
1500	52.85	62.97	-10.12	-16.07%
1750	66.41	66.73	-0.32	-0.48%
2000	66.20	66.43	-0.23	-0.34%
3000	**68.47**	**68.07**	**0.40**	**0.59%**
4000	72.97	71.06	**1.91**	**2.69%**
5000	74.21	71.14	**3.07**	**4.32%**
6000	76.25	73.55	**2.70**	**3.67%**
7000	76.87	74.75	**2.11**	**2.83%**
8000	76.03	73.69	**2.34**	**3.18%**
9000	77.84	77.30	0.54	0.70%
10000	79.21	76.71	2.51	3.27%
12000	78.15	78.27	-0.12	-0.16%
14000	79.99	78.46	1.54	1.96%
16000	79.53	78.37	1.16	1.48%
18000	80.20	78.74	1.46	1.86%

19000	79.98	79.58	0.40	0.51%

Discussion and Conclusion

The primary insight from my experiments is that when the training dataset is smaller than the test dataset, artificially augmenting the training dataset actually hurts the prediction performance. When the training dataset is greater than the validation dataset, applying data augmentation to increase variation improves the model performance. However, the delta between performance with and without augmentation is reduced when the training dataset is sufficiently large.

I find from the only-include-one data augmentation experiments that no specific data augmentation strategy performs significantly better than others. I observe, however, that the brightness level adjustment (dimmer images) even produces diminished accuracy perhaps due to the brightness making it more difficult to differentiate the human face and the background. The results of the ablation study also demonstrates no specific data augmentation strategy affecting the test accuracy more than others. Hyperparameter optimization experiment results show some data augmentation strategies with some parameter values may have slightly better performance than other combinations, but still, the difference (between 0.7993 and 0.8312) is not that significant.

There are several limitations to this study. I only experimented on one dataset, FER2013, which was collected in the wild and already included several variations due to the heterogeneity of the dataset with respect to pose, view angles, and lighting conditions. In future iterations of this study, I hope to perform the same experiments on other datasets with limited variation, for example with datasets collected in the lab-controlled environment such as The Extended Cohn-Kanade Dataset (CK+) [41].

Acknowledgments

I would like to express my sincere gratitude to Dr. Peter Washington for his guidance and mentorship. I would also like to thank my parents for their encouragement and support throughout this research project.

References

[1] Tong, Xiaoyun, Songlin Sun, and Meixia Fu. "Data augmentation and second-order pooling for facial expression recognition." *IEEE Access* 7 (2019): 86821-86828.

[2] Porcu, Simone, Alessandro Floris, and Luigi Atzori. "Evaluation of data augmentation techniques for facial expression recognition systems." *Electronics* 9.11 (2020): 1892.

[3] Zhong, Zhun, et al. "Random erasing data augmentation." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. No. 07. 2020.

[4] Xu, Tian, et al. "Investigating bias and fairness in facial expression recognition." *European Conference on Computer Vision*. Springer, Cham, 2020.

[5] Ahmed, Tawsin Uddin, et al. "Facial expression recognition using convolutional neural network with data augmentation." *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. IEEE, 2019.

[6] Kuo, Chieh-Ming, Shang-Hong Lai, and Michel Sarkis. "A compact deep learning model for robust facial expression recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.

[7] Liu, Ping, et al. "Point adversarial self mining: A simple method for facial expression recognition in the wild." *arXiv preprint arXiv:2008.11401* (2020).

[8] Pitaloka, Diah Anggraeni, et al. "Enhancing CNN with preprocessing stage in automatic emotion recognition." *Procedia computer science* 116 (2017): 523-529.

[9] Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.

[10] Wei, Yunchao, et al. "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[11] Longpre, Shayne, and Ajay Sohmshetty. "Facial keypoint detection." *Facial Detection Kaggle competition* (2016).

[12] Dileep, Prathima, Bharath Kumar Bolla, and Sabeesh Ethiraj. "Revisiting Facial Key Point Detection: An Efficient Approach Using Deep Neural Networks." *arXiv preprint arXiv:2205.07121* (2022).

[13] Agrawal, Abhinav, and Namita Mittal. "Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy." *The Visual Computer* 36.2 (2020): 405-412.

[14] Yang, Huiyuan, Han Yu, and Akane Sano. "Empirical Evaluation of Data Augmentations for Biobehavioral Time Series Data with Deep Learning." *arXiv preprint arXiv:2210.06701* (2022).

[15] Bayer, Markus, et al. "Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers." *International journal of machine learning and cybernetics* (2022): 1-16.

[16] Liu, Zhentao, et al. "A facial expression emotion recognition based human-robot interaction system." IEEE/CAA Journal of Automatica Sinica 4.4 (2017): 668-676.

[17] Spezialetti, Matteo, Giuseppe Placidi, and Silvia Rossi. "Emotion recognition for human-robot interaction: Recent advances and future perspectives." Frontiers in Robotics and AI (2020): 145.

[18] Perez-Gaspar, Luis-Alberto, Santiago-Omar Caballero-Morales, and Felipe Trujillo-Romero. "Multimodal emotion recognition with evolutionary computation for human-robot interaction." Expert Systems with Applications 66 (2016): 42-61.

[19] Deng, Jia, et al. "cGAN based facial expression recognition for human-robot interaction." IEEE Access 7 (2019): 9848-9859.

[20] Chen, Luefeng, et al. "Two-layer fuzzy multiple random forest for speech emotion recognition in human-robot interaction." Information Sciences 509 (2020): 150-163.

[21] Kline, Aaron, et al. "Superpower glass." GetMobile: Mobile Computing and Communications 23.2 (2019): 35-38.

[22] Voss, Catalin, et al. "Superpower glass: delivering unobtrusive real-time social cues in wearable systems." Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct. 2016.

[23] Voss, Catalin, et al. "Effect of wearable digital intervention for improving socialization in children with autism spectrum disorder: a randomized clinical trial." JAMA pediatrics 173.5 (2019): 446-454.

[24] Daniels, Jena, et al. "Exploratory study examining the at-home feasibility of a wearable tool for social-affective learning in children with autism." NPJ digital medicine 1.1 (2018): 32.

[25] Washington, Peter, et al. "A wearable social interaction aid for children with autism." Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. 2016.

[26] Washington, Peter, et al. "SuperpowerGlass: a wearable aid for the at-home therapy of children with autism." Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies 1.3 (2017): 1-22.

[27] Kalantarian, Haik, et al. "A gamified mobile system for crowdsourcing video for autism research." 2018 IEEE international conference on healthcare informatics (ICHI). IEEE, 2018.

[28] Kalantarian, Haik, et al. "The performance of emotion classifiers for children with parent-reported autism: quantitative feasibility study." JMIR mental health 7.4 (2020): e13174.

[29] Penev, Yordan, et al. "A mobile game platform for improving social communication in children with autism: a feasibility study." Applied clinical informatics 12.05 (2021): 1030-1040.

[30] Deveau, Nicholas, et al. "Machine learning models using mobile game play accurately classify children with autism." Intelligence-Based Medicine 6 (2022): 100057.

[31] Chi, Nathan A., et al. "Classifying Autism from Crowdsourced Semi-Structured Speech Recordings: A Machine Learning Approach." arXiv preprint arXiv:2201.00927 (2022).

[32] Zepf, Sebastian, et al. "Driver emotion recognition for intelligent vehicles: A survey." ACM Computing Surveys (CSUR) 53.3 (2020): 1-30.

[33] Lisetti, Christine L., and Fatma Nasoz. "Affective intelligent car interfaces with emotion recognition." Proceedings of 11th International Conference on Human Computer Interaction, Las Vegas, NV, USA. 2005.

[34] Leng, H., Y. Lin, and L. A. Zanzi. "An experimental study on physiological parameters toward driver emotion recognition." Ergonomics and Health Aspects of Work with Computers: International Conference, EHAWC 2007, Held as Part of HCI International 2007, Beijing, China, July 22-27, 2007. Proceedings. Springer Berlin Heidelberg, 2007.

[35] Li, Wenbo, et al. "Cogemonet: A cognitive-feature-augmented driver emotion recognition model for smart cockpit." IEEE Transactions on Computational Social Systems 9.3 (2021): 667-678.

[36] Paikrao, Pavan D., et al. "Smart emotion recognition framework: A secured IOVT perspective." IEEE Consumer Electronics Magazine 12.1 (2021): 80-86.

[37] Katsis, Christos D., et al. "Emotion recognition in car industry." Emotion Recognition: A Pattern Analysis Approach (2015): 515-544.

[38] Xiao, Huafei, et al. "On-road driver emotion recognition using facial expression." Applied Sciences 12.2 (2022): 807.

[39] Van Dyk, David A., and Xiao-Li Meng. "The art of data augmentation." Journal of Computational and Graphical Statistics 10.1 (2001): 1-50.

[40] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of big data 6.1 (2019): 1-48.

[41] Lucey, Patrick, et al. "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression." 2010 ieee computer society conference on computer vision and pattern recognition-workshops. IEEE, 2010.