Data Augmentation for image classification and applying grad-cam with small dogs and cats dataset

Data augmentation techniques are used in nearly every state-of-the-art machine learning model in applications such as image and text classification. Modern machine learning models, such as deep neural networks, can have billions of parameters and require large labeled training datasets—which are often unavailable. Augmenting an existing dataset in a more cost-effective way than collecting additional data. In the case of image classification applications, data enhancement is usually performed using geometric transformation techniques applied to the original images, which we will discuss in more detail below. In this report, we will apply data enhancement methods to improve the performance of the classification model on the small dogs and cats dataset. After augmented data and input to the classification model, I would suggest using a popular backbone network model - VGG16 to extract features and conduct classification.

1. Contents and Methods

1.1 VGG16

A VGG network consists of small convolution filters. VGG16 has three fully connected layers and 13 convolutional layers (**figure 1**).

In this report, I have changed the input to the network is an image of dimensions (128, 128, 3).

Convolutional layers—the convolutional filters of VGG use the smallest possible receptive field of 3×3. VGG also uses a 1×1 convolution filter as the input's linear transformation.

ReLu activation—next is the Rectified Linear Unit Activation Function (ReLU) component, AlexNet's major innovation for reducing training time. ReLU is a linear function that provides a matching output for positive inputs and outputs zero for negative inputs. VGG has a set convolution stride of 1 pixel to preserve the spatial resolution after convolution (the stride value reflects how many pixels the filter "moves" to cover the entire space of the image).

Hidden layers - all the VGG network's hidden layers use ReLU instead of Local Response Normalization like AlexNet. The latter increases training time and memory consumption with little improvement to overall accuracy.

Pooling layers–A pooling layer follows several convolutional layers—this helps reduce the dimensionality and the number of parameters of the feature maps created by each convolution step. Pooling is crucial given the rapid growth of the number of available filters from 64 to 128, 256, and eventually 512 in the final layers.

Fully connected layers—VGGNet includes three fully connected layers. The first two layers each have 4096 channels, and the third layer has 2 channels, one for every class of dogs and cats.

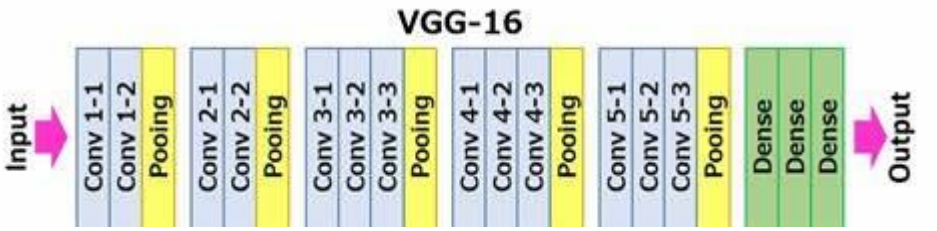

Figure 1. VGG-16 architecture

1.2 Data Augumentation

1.2.1 Mixup Augumentation

In this approach, the two images will be linearly mixed and their labels in the same proportion [1]. The model will be able to see a different image with different label during training. It enables the model to make smooth decision boundaries while classifying the object since the model using linearly interpolated images and labels for classification decision intead of binary decision. The mixup of images and labels, the new image will have the labels [0.19, 0.81] which means the class distribution is tilted more towards dog and the mixup visualization also proves this (figure 2**)**.

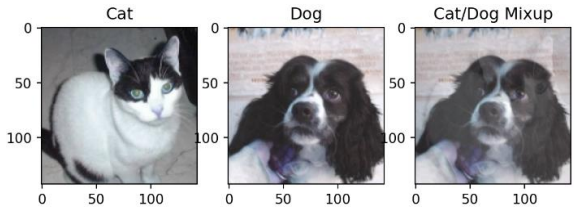

Figure 2. The mixup of dog and cat images and labels

1.2.2 Cutout Augumentation

The cutout method [2] is to simulate the situation of object occlusion that is mostly encountered in tasks such as object recognition or human pose estimation. Instead of model seeing the same image everytime, it sees different parts of that image which helps it to perform well. The occluded part does not contain any information. In the image as figure 3, the randomly occluded part is replaced by all 0s.

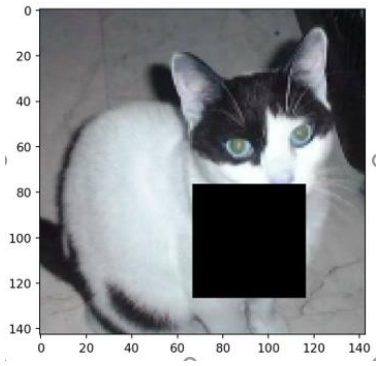

Figure 3. The cutout of a cat image

1.2.3 Cutmix Augumentation

In cutout, the occluded part does not contain any information which in unwanted since it just increases resource usage without adding any value. cutmix utilizes above two techniques to mix the images and utilize the occluded part [3]. In the image as figure 4, the cutout portion in dog image is replaced by the same portion of cat image. It also linearly interpolates the label as in mixup. As seen in the example below, the new labels are [0.18, 0.82] for cat and dog respectively and this can be verified from image as well.

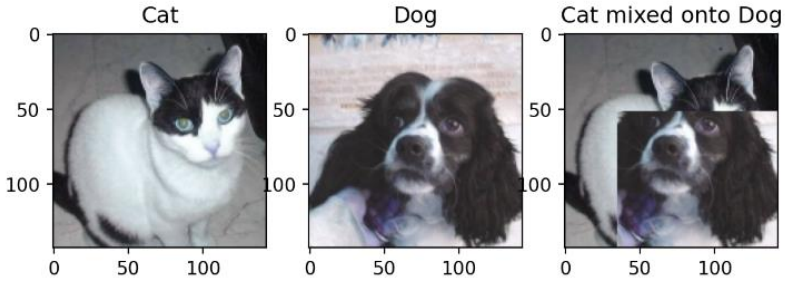

Figure 4. The cutmix of dog and cat images and labels

1.2.4 Label smoothing

In this report, method applied to implement label smoothing utilizes Keras/TensorFlow's CategoricalCrossentropy class directly.

The benefit here is that don't need to implement any custom function — label smoothing can be applied on the fly when instantiating the BinaryCrossentropy class with the label_smoothing parameter, like so: BinaryCrossentropy(label_smoothing=0.1) [4]

1.3 Grad-Cam

Grad-CAM works by finding the final convolutional layer in the network and then examining the gradient information flowing into that layer [5].

The output of Grad-CAM is a heatmap visualization for a given class label (either the top, predicted label or an arbitrary label). This heatmap's benefit is to visually verify where in the image the CNN is looking.

1.4 AUC

AUC is just the area underneath the entire ROC curve. AUC provides us with a nice, single measure of performance for classifiers, independent of the exact classification threshold chosen. This allows to compare models to each other without even looking at their ROC curves. AUC ranges in value from 0 to 1, with higher numbers indicating better performance. A perfect classifier will have an AUC of 1, while a perfectly random classifier an AUC of 0.5. A model that always predicts that a negative sample is more likely to have a positive label than a positive sample will have AUC of 0, indicating a severe failure on the modeling side. Scores in the range [0.5, 1] imply good performance, while anything under 0.5 indicates very poor performance.

2. Training model

This model deployed on google collab and using two methods of cutmix and mixup augmentation gives better results than cutout augmentation. It shows in the below table.

Backbone Network	ImageNet Cls Top-1 Error (%)	Detection		Image Captioning	
		SSD [24] (mAP)	Faster-RCNN [30] (mAP)	NIC [43] (BLEU-1)	NIC [43] (BLEU-4)
ResNet-50 (Baseline)	23.68	76.7 (+0.0)	75.6 (+0.0)	61.4 (+0.0)	22.9 (+0.0)
Mixup-trained	22.58	76.6 (-0.1)	73.9 (-1.7)	61.6 (+0.2)	23.2 (+0.3)
Cutout-trained	22.93	76.8 (+0.1)	75.0 (-0.6)	63.0 (+1.6)	24.0 (+1.1)
CutMix-trained	21.40	**77.6** (+0.9)	**76.7** (+1.1)	**64.2** (+2.8)	**24.9** (+2.0)

Table 1. The results show the data augmentation methods in the trained model. [3]

2.1 Cutmix

```
Epoch 298/300
63/63 [==============================] - 10s 156ms/step - loss: 0.4952 - accuracy: 0.8695 - val_loss: 0.5754 - val_accuracy: 0.7290
Epoch 299/300
63/63 [==============================] - 10s 156ms/step - loss: 0.4883 - accuracy: 0.8715 - val_loss: 0.5817 - val_accuracy: 0.7160
Epoch 300/300
63/63 [==============================] - 10s 155ms/step - loss: 0.4948 - accuracy: 0.8700 - val_loss: 0.5764 - val_accuracy: 0.7250
32/32 [==============================] - 1s 44ms/step
32/32 [==============================] - 1s 44ms/step
[INFO] serializing network...
              precision    recall  f1-score   support

           0       0.72      0.69      0.70       500
           1       0.70      0.73      0.71       500

    accuracy                           0.71      1000
   macro avg       0.71      0.71      0.71      1000
weighted avg       0.71      0.71      0.71      1000
```

Figure 5. The trained model shows the results of accuracy, loss when applying cutmix augmentation

Data augmentation is only performed on the training set because it makes the model more general and robust. Therefore, it is not necessary to augment the valid and test set. The network trained for 300 epochs and it achieved high accuracy, 87% classification accuracy on the training set and 72.50% accuracy on the validating set.

The average loss that follows the training loss, as is apparent from the plot below (figure 6):

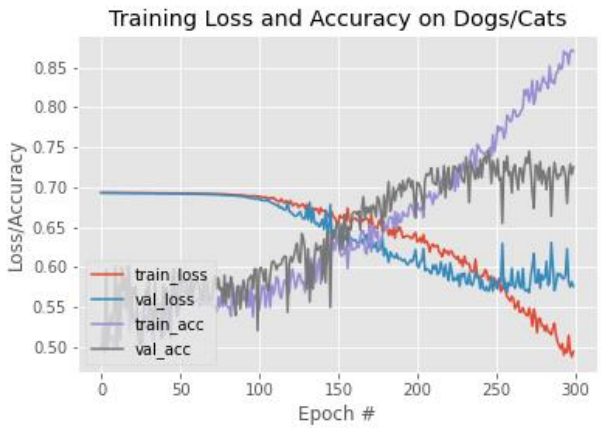

Figure 6. The graph shows the whole process of accuracy and loss in dogs and cats.

In the image below that the area under the red and curve (AUC) of dogs and cats is equal to 0.795. At 0.795, model's AUC trained isn't too shabby (figure 7).

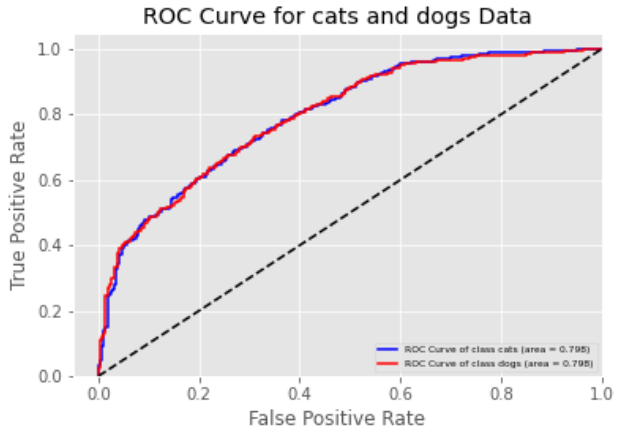

Figure 7. The chart provides information about ROC Curve and AUC

2.2 Mixup

```
Epoch 298/300
63/63 [==============================] - 10s 155ms/step - loss: 0.2992 - accuracy: 0.9835 - val_loss: 0.6776 - val_accuracy: 0.6540
Epoch 299/300
63/63 [==============================] - 10s 156ms/step - loss: 0.2942 - accuracy: 0.9840 - val_loss: 0.6785 - val_accuracy: 0.6520
Epoch 300/300
63/63 [==============================] - 10s 155ms/step - loss: 0.3030 - accuracy: 0.9830 - val_loss: 0.6794 - val_accuracy: 0.6470
32/32 [==============================] - 1s 44ms/step
32/32 [==============================] - 1s 44ms/step
[INFO] serializing network...
              precision    recall  f1-score   support

           0       0.62      0.70      0.66       500
           1       0.66      0.58      0.61       500

    accuracy                           0.64      1000
   macro avg       0.64      0.64      0.64      1000
weighted avg       0.64      0.64      0.64      1000
```

Figure 8. The trained model shows the results of accuracy, loss when applying Mixup augmentation

In the mixup augmentation, the model achieved high accuracy, 98.30% classification accuracy on the training set and 64.70% accuracy on the validating set. The classification accuracy on the testing set with about 64% which is lower than the mixup augmentation of 71%. The average loss that follows the training loss, as is apparent from the plot below:

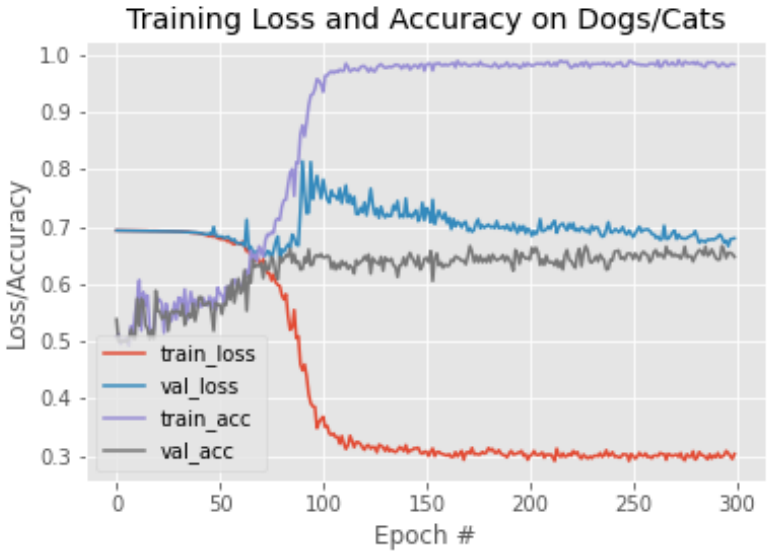

Figure 9. The graph shows the whole process of accuracy and loss in dogs and cats.

In the figure below, the area under the curve and red (AUC) of the dog and cat is equal to 0.71. This number is lower than the cutmix augmentation of 0.795. Thereby it is found that when applying increased cutmix, the performance will be higher.

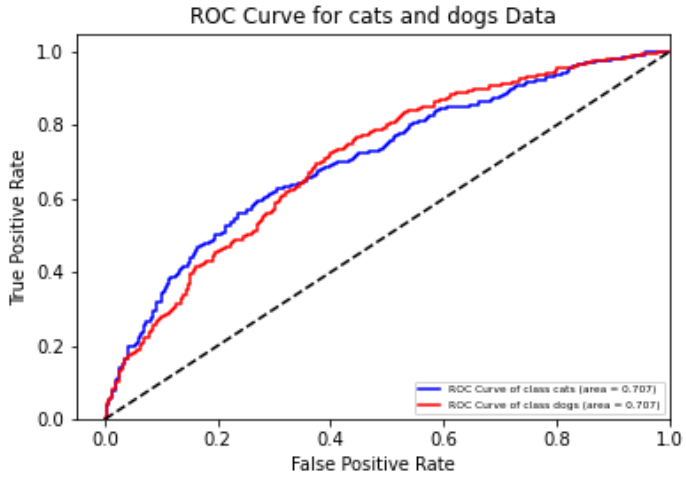

Figure 10. The chart provides information about ROC Curve and AUC

3. Results

With the small dog and cat dataset, CutMix enhances the classification performance by +0.07 (7% accuracy), outperforming the Mixup performance. Despite the result of the trained model when not applying data augmentation with the accuracy of 0.77 and AUC 0.832 is quite high, the loss value of valid data is very high at about 1.7147. These show unable to accurately model the training data hence generating large errors and the model is underfitting. Also, the AUC value of 0.71 in the mixup augmentation is higher than the cutmix augmentation value of 0.795 after 300 epochs.

Table 2. Comparison of accuracy and AUC results between mixup, cutmix, and not augmented

	Accuracy	**Auc**
VGG16 (Not augmented)	0.77	0.832
Mixup	0.64	0.71
Cutmix	0.71	0.795

Show the Grad-CAM map on top of the below image. Clearly, the upper face and ear of the dog have the greatest impact on the classification. The densely displayed map on 2 images of the cutmix method shows the effect that this method brings.

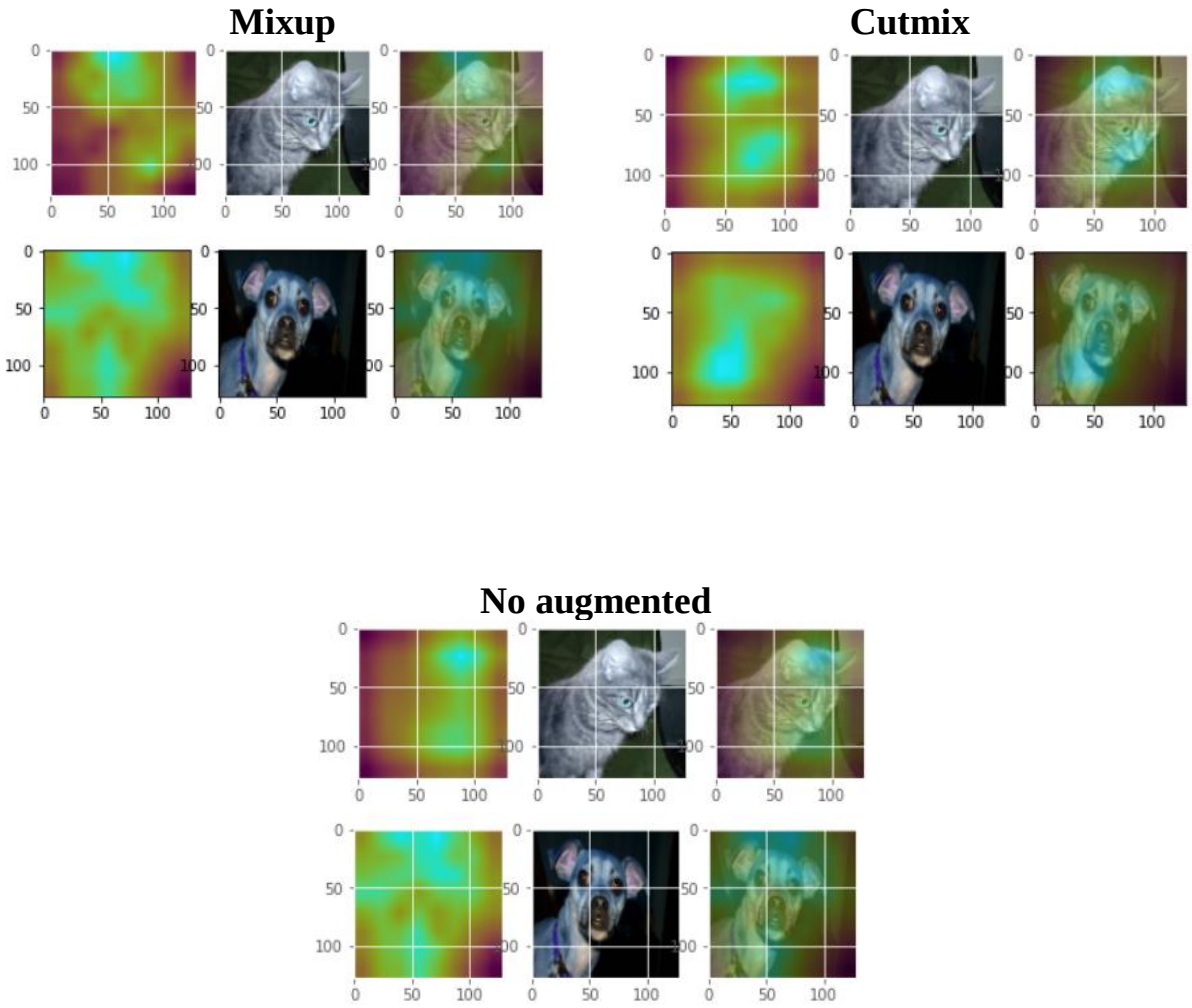

Figure 11. The images show the results when applying grad-cam between mixup, cutmix, and not augmented

Although the main purpose of this model is used to classify dogs and cats. In the small dog and cat dataset, it appears with some human faces. To test and observe this model, a human face will be examined and tested in the cutmix augmentation method. The results are illustrated in figure 12.

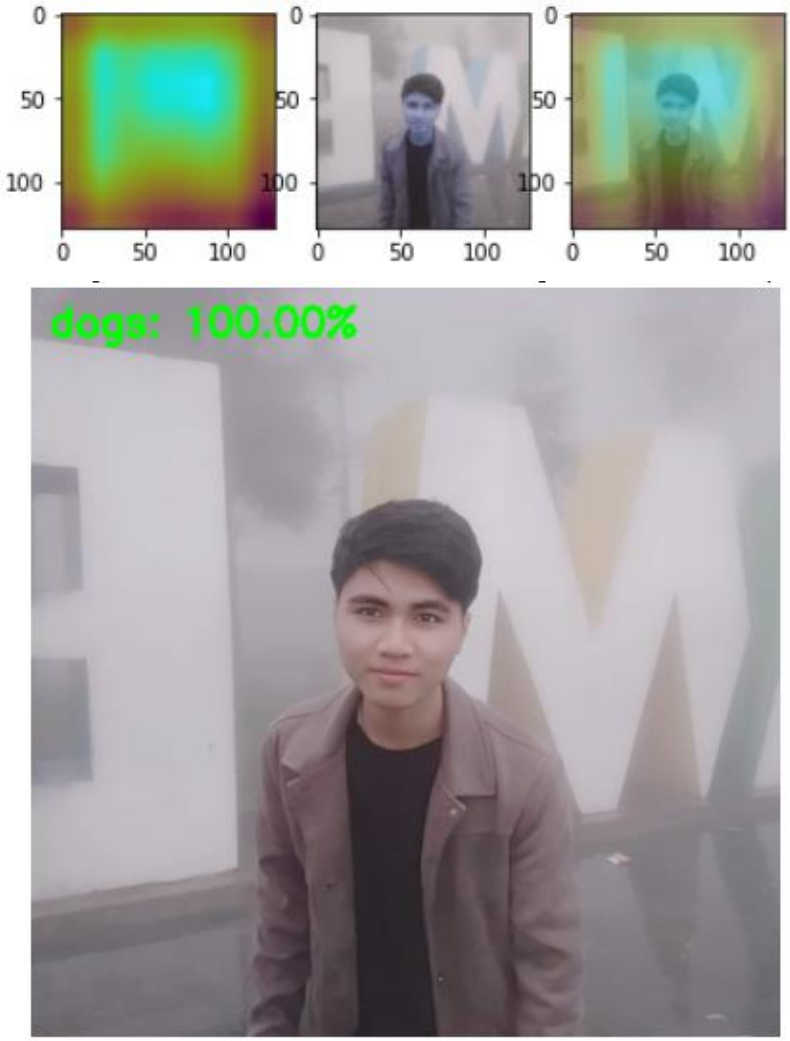

Figure 12. Test model with a human face

It is clear that the results of a human face lean towards the dog label with 100% accuracy. Because, with a range of similarities between human and dog face processing. Both dog and human negative and positive expressions were recognized from both full and cropped faces. These results extend existing work on cross-species similarities in facial emotions and provide evidence that these similarities are naturally exploited when humans interact with dogs (as figure 13).

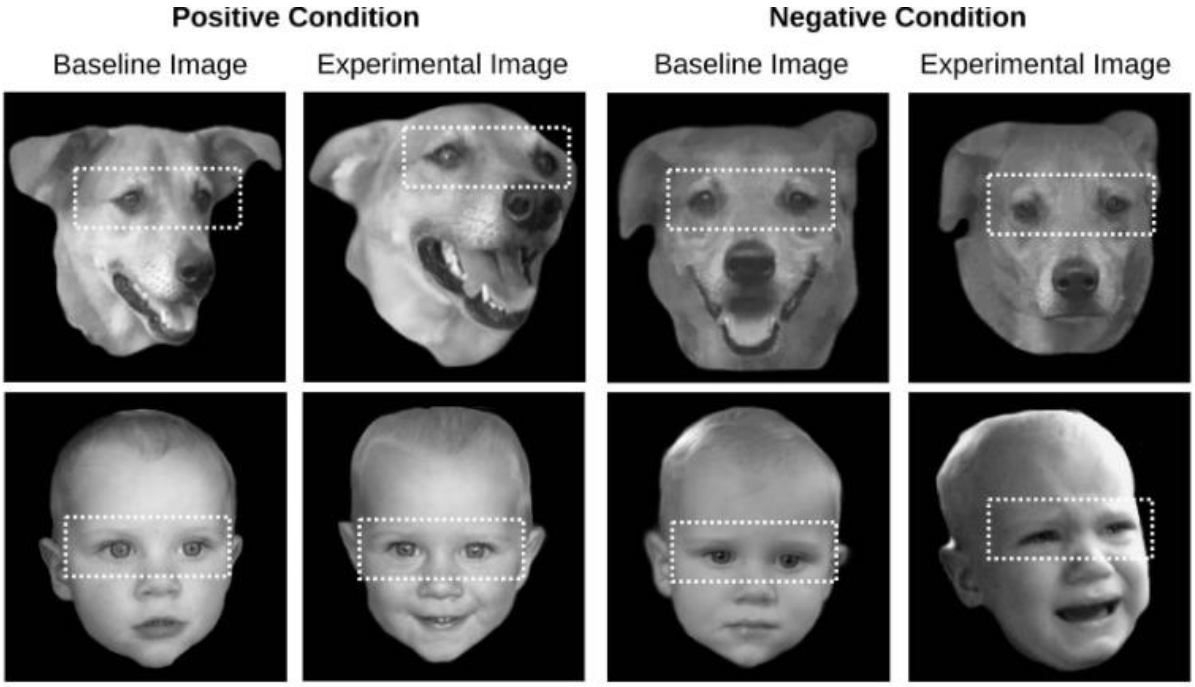

Figure 13. Exemplary images for the positive and negative conditions of a human and dog [6]

4. References

1. Zhang, Hongyi et al. “mixup: Beyond Empirical Risk Minimization.” arXiv:1710.09412**.**
2. Devries, Terrance and Graham W. Taylor. “Improved Regularization of Convolutional Neural Networks with Cutout.” arXiv:1708.04552**.**
3. S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo and J. Choe, "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 6022-6031, doi: 10.1109/ICCV.2019.00612.
4. https://keras.io/api/losses/probabilistic_losses/#:~:text=tf.keras.losses.Binary Crossentropy
5. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.
6. Schirmer, Annett et al. “Humans Process Dog and Human Facial Affect in Similar Ways.” *PLoS ONE* 8 (2013).