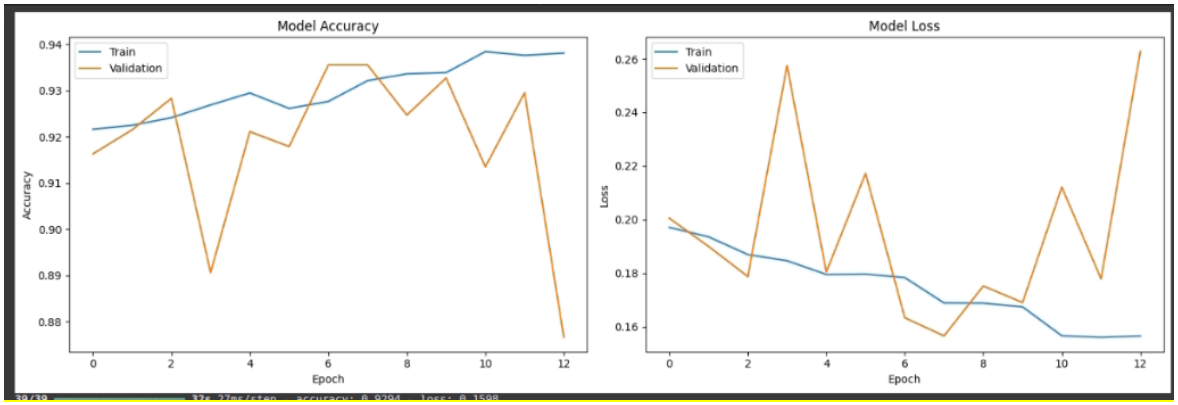
LAB6

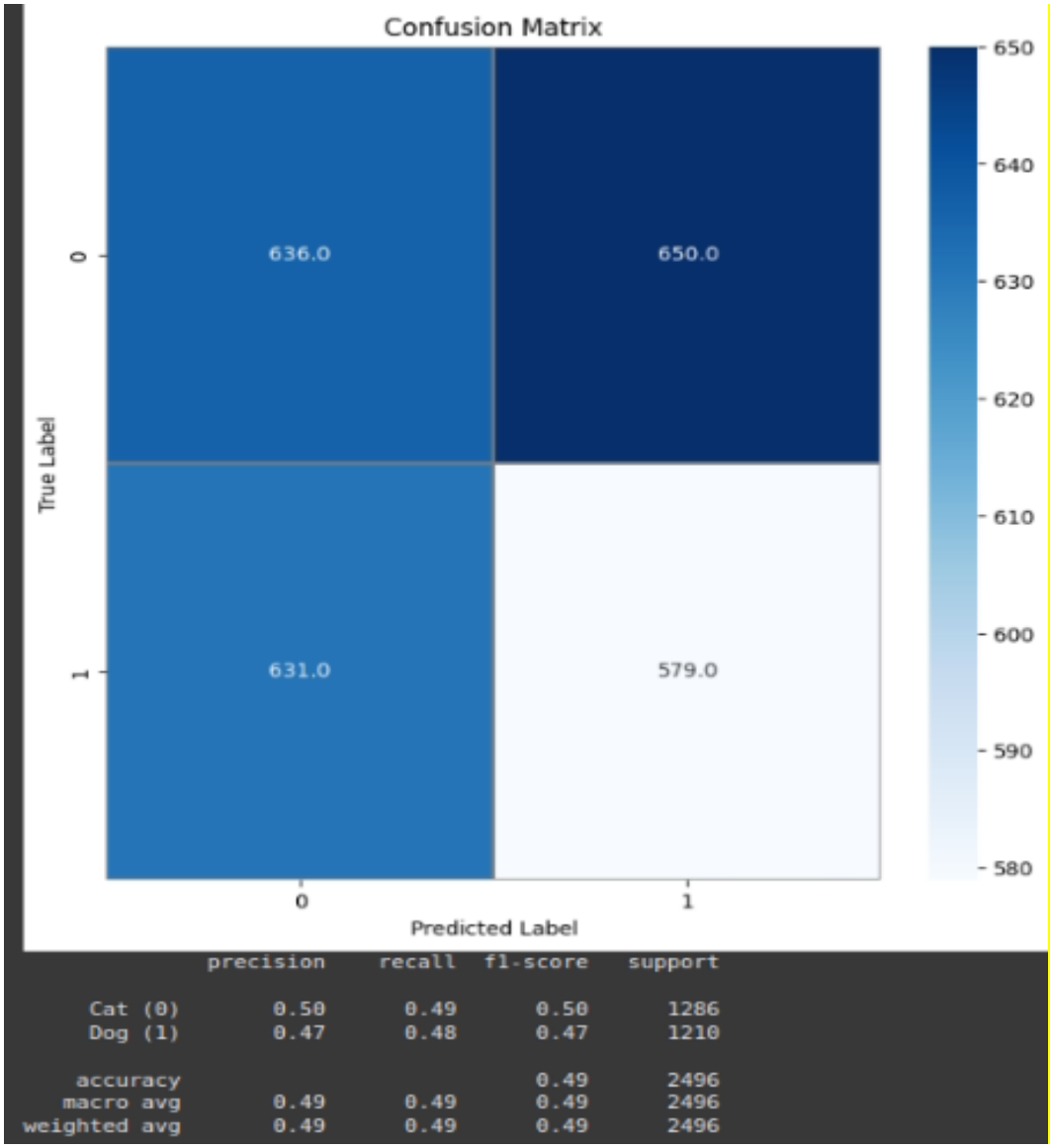
Results:

Model: "sequential"

Layer (type)	Output Shape	Param #
random_flip (RandomFlip)	(None, 128, 128, 3)	0
random_rotation (RandomRotation)	(None, 128, 128, 3)	0
random_zoom (RandomZoom)	(None, 128, 128, 3)	0
conv2d (Conv2D)	(None, 128, 128, 32)	896
batch_normalization (BatchNormalization)	(None, 128, 128, 32)	128
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_1 (Conv2D)	(None, 64, 64, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_2 (Conv2D)	(None, 32, 32, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 32, 32, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 128)	0
dropout (Dropout)	(None, 16, 16, 128)	0
conv2d_3 (Conv2D)	(None, 16, 16, 256)	295,168
batch_normalization_3 (BatchNormalization)	(None, 16, 16, 256)	1,024
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 256)	0
dropout_1 (Dropout)	(None, 8, 8, 256)	0
flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 256)	4,194,560
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

Total params: 4,585,153 (17.49 MB)
Trainable params: 4,584,193 (17.49 MB)
Non-trainable params: 960 (3.75 KB)





	precision	recall	f1-score	support
Cat (0)	0.50	0.49	0.50	1286
Dog (1)	0.47	0.48	0.47	1210
accuracy			0.49	2496
macro avg	0.49	0.49	0.49	2496
weighted avg	0.49	0.49	0.49	2496

Q:What changes did you make to the CNN architecture? Why did you choose those modifications, and how did they affect model performance?

A:I added RandomFlip, RandomRotation, and RandomZoom to the CNN architecture to increase data diversity and make the model more adaptable to different image changes.

The model depth is increased to four convolutional blocks (filter number 32 → 64 → 128 → 256) to capture hierarchical features from simple edges to complex patterns.
Each convolution is followed by Batch Normalization to stabilize the training, and gradually increasing Dropout (0.3 → 0.4 → 0.5) is added to the deep layers to suppress overfitting.

The optimizer uses Adam and adjusts the learning rate to 0.0005.

These modifications did make the training process more stable, but judging from the test set results, the model still overfitted. It is speculated that the data features are too simple or the amount of data is insufficient, resulting in poor generalization ability.

Q:What methods did you use to reduce overfitting (if any)? How effective were they? Explain with reference to training/validation curves.

A:I used Data Augmentation, Dropout, Batch Normalization, Early Stopping, and Model Checkpoint techniques to reduce overfitting.
These methods make the loss continue to decrease and the accuracy steadily increase on the training set, but the validation loss fluctuates significantly, indicating that the generalization performance of the model on different data is still unstable.
Overall, these methods "mitigated overfitting but did not completely solve it."

Q:Based on your confusion matrix, what kinds of misclassifications occurred most frequently? What might be causing these errors? How would you attempt to reduce these errors?

A:According to the confusion matrix, the most common errors are misclassifying cats as dogs (650 times) and misclassifying dogs as cats (631 times). The two errors are close in number, indicating that the model does not have a good grasp of the difference between cats and dogs.

Possible causes of the error include:

Insufficient feature extraction, unable to extract discriminative features

The dataset may have many cats and dogs that look similar (e.g., angle, background, breed).

The model capacity is large, but it is not effectively utilized

To reduce these errors, I would consider:

Enhanced data enhancement, such as larger angle rotation and affine transformation

Use models that are more suitable for small sample classification, such as MobileNet, EfficientNet

Fine-tune learning rate schedule, add CosineAnnealing / ReduceLROnPlateau

Use feature extractors or pre-trained backbones to improve feature recognition capabilities