Summer Course Group 1:

- Fikry Idham Dwiyana (Group Leader)
- Ananda Myzza Marhelio

Assignment - Course 4B

Model Architecture

Three Convolutional Neural Network (CNN) architectures were tested, each progressively integrating batch normalization, dropout, and data augmentation to evaluate their impact on model performance.

- **First Model**: Basic CNN architecture.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
flatten (Flatten)	(None, 82944)	0
dense (Dense)	(None, 64)	5,308,480
dense_1 (Dense)	(None, 16)	1,040
dense_2 (Dense)	(None, 10)	170

Total params: 15,987,248 (60.99 MB)

Trainable params: 5,329,082 (20.33 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 10,658,166 (40.66 MB)

- **Second Model**: Added batch normalization.

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 148, 148, 32)	896
batch_normalization (BatchNormalization)	(None, 148, 148, 32)	128
max_pooling2d_2 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_3 (Conv2D)	(None, 72, 72, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 72, 72, 64)	256
max_pooling2d_3 (MaxPooling2D)	(None, 36, 36, 64)	0
flatten_1 (Flatten)	(None, 82944)	0
dense_3 (Dense)	(None, 64)	5,308,480
dense_4 (Dense)	(None, 16)	1,040
dense_5 (Dense)	(None, 10)	170

Total params: 15,988,016 (60.99 MB)

Trainable params: 5,329,274 (20.33 MB)

Non-trainable params: 192 (768.00 B)

Optimizer params: 10,658,550 (40.66 MB)

- **Third Model**: Added dropout.

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 148, 148, 32)	896
batch_normalization_2 (BatchNormalization)	(None, 148, 148, 32)	128
max_pooling2d_4 (MaxPooling2D)	(None, 74, 74, 32)	0
dropout (Dropout)	(None, 74, 74, 32)	0
conv2d_5 (Conv2D)	(None, 72, 72, 64)	18,496
batch_normalization_3 (BatchNormalization)	(None, 72, 72, 64)	256
max_pooling2d_5 (MaxPooling2D)	(None, 36, 36, 64)	0
dropout_1 (Dropout)	(None, 36, 36, 64)	0
flatten_2 (Flatten)	(None, 82944)	0
dense_6 (Dense)	(None, 64)	5,308,480
dense_7 (Dense)	(None, 16)	1,040
dropout_2 (Dropout)	(None, 16)	0
dense_8 (Dense)	(None, 10)	170

Total params: 15,988,016 (60.99 MB)

Trainable params: 5,329,274 (20.33 MB)

Non-trainable params: 192 (768.00 B)

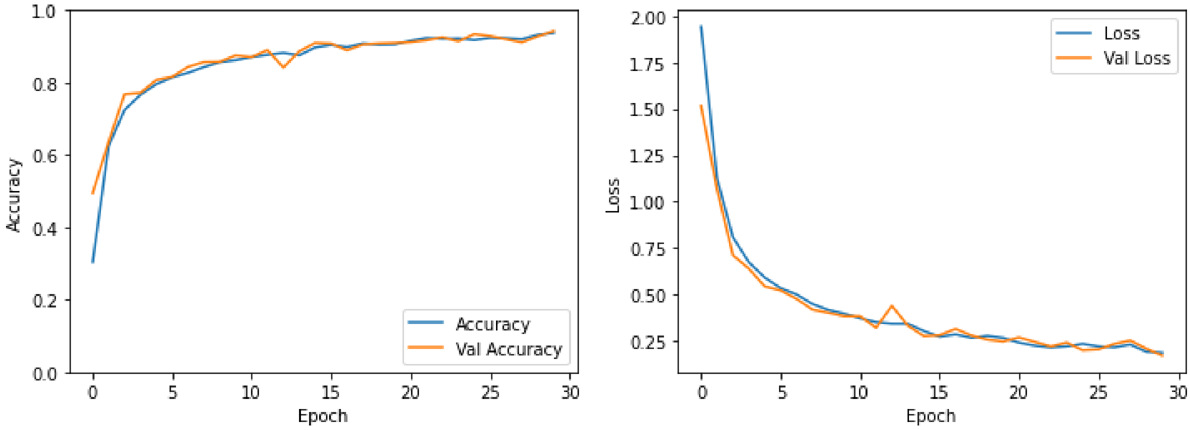
Optimizer params: 10,658,550 (40.66 MB)

Techniques Used

- **Batch Normalization**: Applied after the convolutional layers to normalize the output and improve the learning rate.
- **Dropout**: Used to prevent overfitting by randomly setting a fraction of input units to 0 at each update during training.
- **Data Augmentation**: Applied to all models using the `ImageDataGenerator` with rescaling, rotation, width/height shift, shear, zoom, horizontal flip, and fill mode to increase the diversity of the training data and improve generalization.

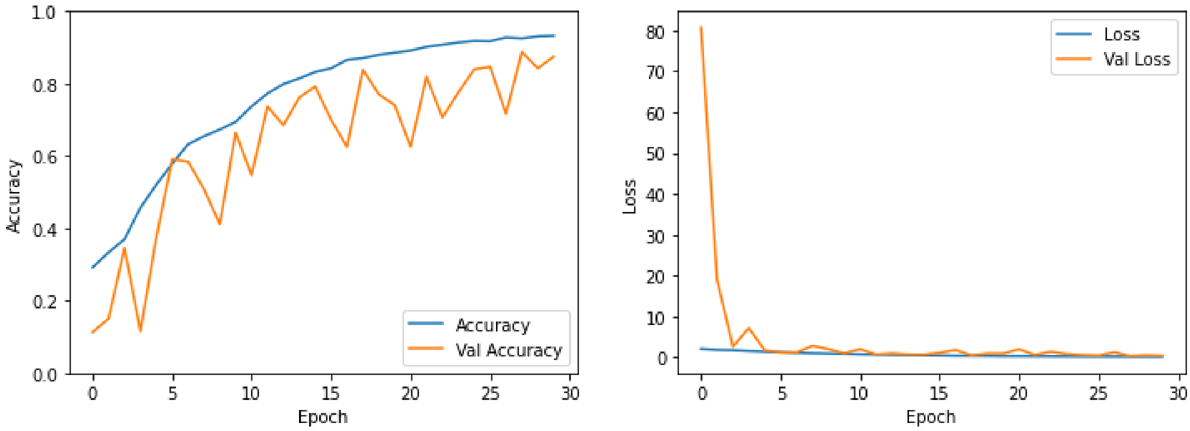
Results

First Model:



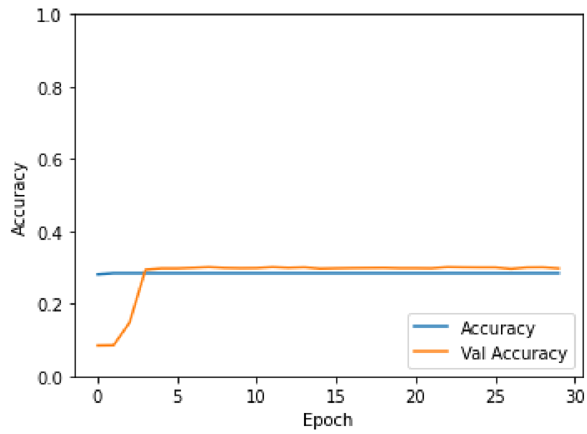
- **Accuracy**: Training: 93.17%, Validation: 94.25%
- **Loss**: Training: 0.2035, Validation: 0.1718

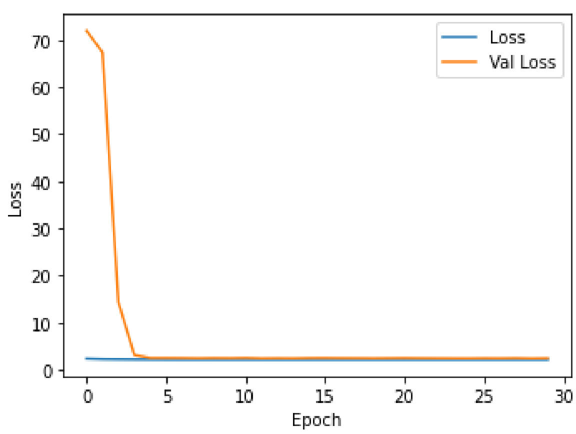
Second Model:



- **Accuracy**: Training: 93.07%, Validation: 87.45%
- **Loss**: Training: 0.2143, Validation: 0.3932

Third Model:





- **Accuracy**: Training: 28.71%, Validation: 29.92%
- **Loss**: Training: 2.1421, Validation: 2.4303

Impact of Techniques

- **Batch Normalization**: The second model with batch normalization demonstrated an initial improvement in stability and convergence compared to the first model. However, it showed inconsistent validation performance and higher loss, likely due to inadequate handling of model complexity and regularization.
- **Dropout**: The third model incorporated dropout but failed to achieve significant improvements. The training and validation performance were poor, indicating either an over-regularization or improper learning rate adjustments. Dropout at 0.25 and 0.5 might have excessively reduced the effective capacity of the model.
- **Data Augmentation**: Data augmentation was implemented across all the models used and was able to make the first model achieve very good results in terms of both training and validation accuracy. Unfortunately, however, the third model's failure suggests that the augmentation techniques were either not sufficient or that they were improperly integrated with the regularization methods, causing an imbalance.

Conclusion

The first model provided the best performance, with high training and validation accuracy, suggesting that simpler architectures might benefit more from careful tuning without excessive regularization. Batch normalization alone improved training stability but didn't translate to better validation performance. The addition of dropout in the third model did not yield the expected improvements, likely due to over-regularization and misconfiguration. Future improvements could include fine-tuning dropout rates and enhancing augmentation techniques to balance regularization and data diversity effectively.