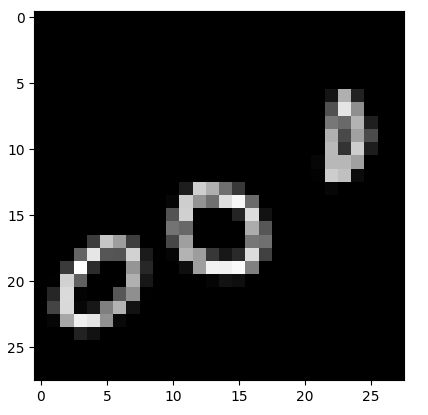
## Task 2

### Dataset Overview and Preprocessing

Task 2 assignment is focusing on the objectives and importance of multi-label classification on multiple digits from images, where each image contains several digits.The dataset for this project consists of images each labeled with three digits.



### Data Loading and Initial Exploration

To load and explore the dataset, we define the load\_images\_from\_folder function, which reads images from subdirectories, resizes them, converts them to grayscale, and assigns labels based on subdirectory names. Directories for training, validation, and test datasets are specified to organize the data. We print the contents of these directories to verify their structure. We load the datasets into arrays of images and corresponding labels. This process prepares the dataset for further processing and model training.

### Preprocessing Steps:

Preprocessing steps taken to prepare the data for modeling. This includes reshaping and normalizing pixel values, one-hot encoding of labels to convert labels to categorical, creation of a custom data generator for efficient data handling, set up learning rate scheduler to adjust the learning rate during training, and configure callbacks for early stopping to prevent overfitting by stop the training when the validation loss stops improving.

### Challenges:

The challenges during preprocessing involved reshaping the images to fit the model's input requirements, normalizing pixel values to the rnge [0, 1], and encoding the labels for a multi-label setup using one-hot encoding. We faced several challenges when we need to choose which activation function and which loss function we need to use.

### Methodology and Techniques:

CNN models were developed, comparing Categorical Crossentropy with Binary Crossentropy as loss functions and Sigmoid versus Softmax as activation functions. Techniques like early stopping, Drop out, were used to improve training performance and avoid overfitting.

### Model Comparison and Selection

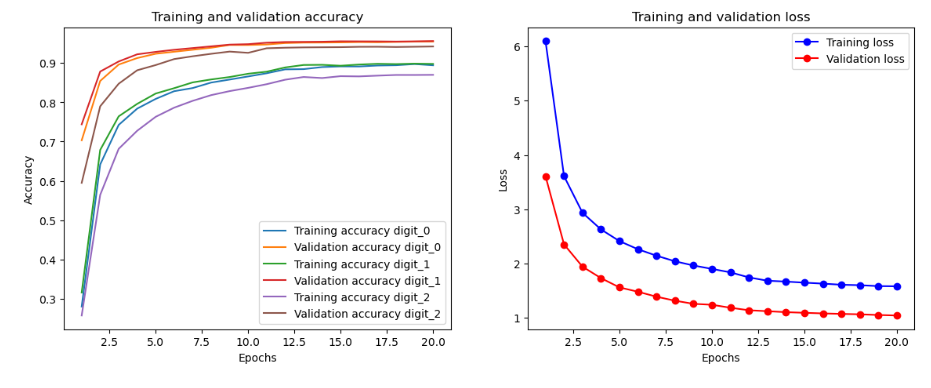
Binary crossentropy as the loss function outperformed categorical crossentropy, showing lower loss and higher accuracy for multi-label classification, demonstrating its suitability for this task. While recommendations suggest Sigmoid for multi-label and Softmax for multi-class, our experiment shows the use of Softmax in output layer achieved the highest accuracy.

### Hyperparameter Tuning

Hyperparameter tuning perfomed using Keras-Tuner to find the best parameters, beside adjust the model parameters like learning rate and increase K-fold cross-validation to more accurately evaluate model performance across different subsets of data, and implement batch normalization in the model to normalize the inputs for each mini-batch, reducing internal covariate shift and stabilizing training.

### Results and Discussion:

Initial models using **Sigmoid activation and categorical crossentropy** as the loss function achieved high accuracy (around 95%) and loss (around 1.00), indicating no underfitting or overfitting.



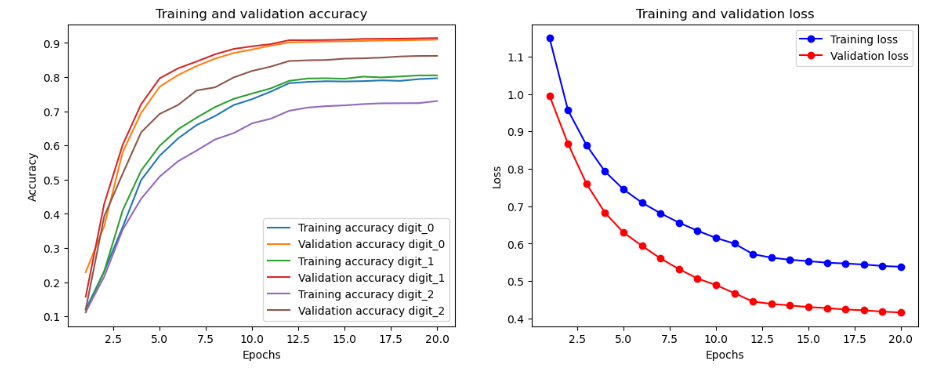
Test loss: 1.0049712657928467

Test accuracy for digit\_0: 0.9526000022888184

Test accuracy for digit\_1: 0.9549999833106995

Test accuracy for digit\_2: 0.9416499733924866

Switching to **Sigmoid activation and binary crossentropy** as the loss function reduced losses (around 0.3), but lowered accuracy. The final evaluation on the test dataset showed accuracies (above 90%) for all digits.



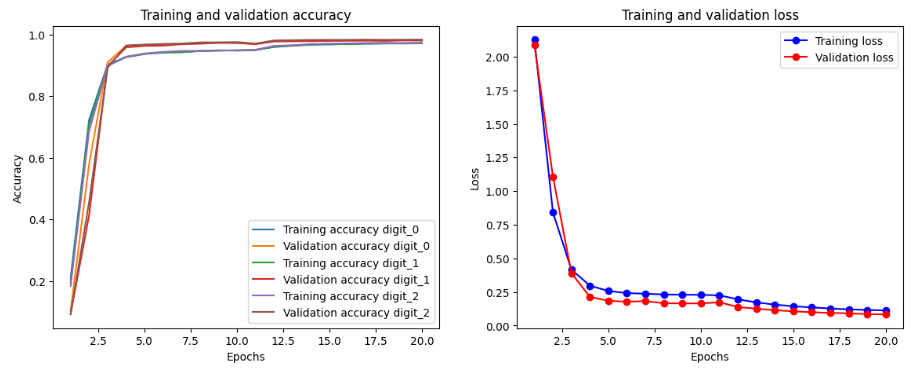
Test loss: 0.3763981759548187

Test accuracy for digit\_0: 0.907800018787384

Test accuracy for digit\_1: 0.9002500176429749

Test accuracy for digit\_2: 0.9164999723434448

The model using **Softmax activation and binary crossentropy** as the loss function showed better performance with higher accuracy (around 98%) and loss (around 0.07).



After hyperparameter tuning, the model demonstrated improvement and better performance.

Using **Keras-Tuner**, we found the best parameters and implemented into the model:

val\_loss: 0.07059618085622787

Best val\_loss So Far: 0.06173119693994522

Total elapsed time: 05h 47m 46s

{'conv\_1\_filters': 96, 'conv\_2\_filters': 128, 'conv\_3\_filters': 128, 'dense\_units': 256}

Final testing on unseen data after Hyperparameter Tuning on Softmax activation function and Binary Crossentropy as the loss function confirmed the model's highest accuracy (around 98.7%) and lowest loss (around 0.007).

Test loss: 0.06563404202461243

0.008683374151587486

0.008130848407745361

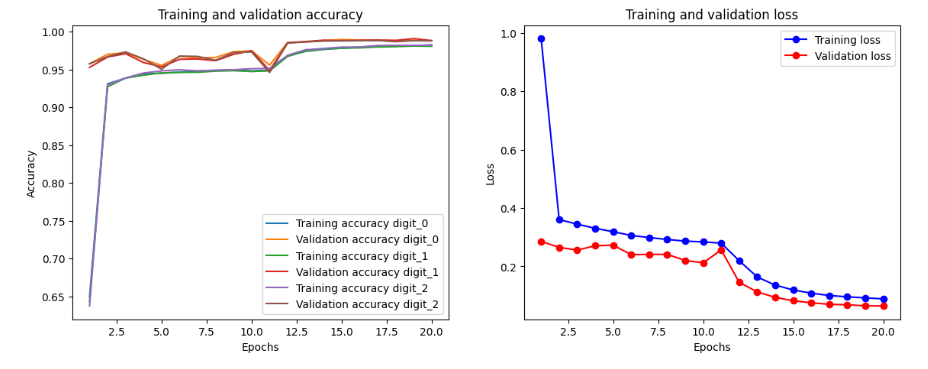
0.007283779326826334

0.986299991607666

0.9872499704360962

0.9888499975204468

There is high accuracy on the test set which the model has never seen before, that means the model can predict on the new unseen data. The model regularized well and trained appropriately, no overfitting nor underfitting.



### Visualization

### Confusion Matrix

### 

A graph of blue squares

Description automatically generated

A graph of a diagram

Description automatically generated with medium confidence

The diagonal elements represent the correctly classified ones, with the highest values along the diagonal showing correct predictions. Off-diagonal values show where the model making errors.

### The model is confused when distinguishing between digits 0 and 3, and also between digits 0 and 8. Their features are similar to each other and are not distinguished well by the model.

### Visual Representation of The Filters/ Kernels and Feature map

Filters/Kernel is small matrices that slide over the input image to detect specific features such as edges, textures, or patterns. Each filter is applied across the entire input image, and the result is a feature map.

The visual representation of the filters helps to understand what each filter is focusing on in the input images. The filters have dimensions (3, 3, 1, 96), where 3 and 3 are the height and width of each filter. 1 is the number of input channels for grayscale image, there is only one channel). 96 is the number of filters in this layer.

The filters have purpose to detect specific features within the input images. During training, the weights of these filters are adjusted to minimize the loss function, thereby learning the features that are most relevant for the task. Visualizing the filters can give insights into what the model is learning.

A grid of squares with different shapes

Description automatically generated with medium confidence

A grid of squares with small white dots

Description automatically generated

A grid of squares with different shades of black and white squares

Description automatically generated

The patterns seen in the feature show of what each filter has learned. For example, some may detect edges, other textures, and some specific shapes. Early layers earn simple features , then deeper layers learn more complex features and patterns.

### Conclusion

The project successfully demonstrated the application of CNNs to multi-label image-based digit classification, with Softmax activation and binary crossentropy as the loss function showing the highest performance over Sigmoid activation and categorical crossentropy as loss function. They are more improving after hyperparameter tuning. This is showing how importance in selecting appropriate loss functions and optimizing model architectures.

Future work could explore more complex architectures, additional regularization techniques, and further hyperparameter tuning to enhance model performance and robustness, potentially experimenting with different convolutional layer configurations to improve focus on relevant parts of the image.

### Ethical, Legal and Social Considerations in AI

When developing AI models for multi-label image classification, such as with triple MNIST digits, it's important to address ethical, legal, and social issues.

Data protection laws like GDPR or CCPA must be adhered to, ensuring that any personal data is handled with strict privacy measures.

The ownership of the AI model, its training data, and its outputs need to carefully consider ensuring responsible and beneficial AI deployment.