I used the Keras Sequential API, where you have just to add one layer at a time, starting from the input.

The first is the convolutional (Conv2D) layer. It is like a set of learnable filters. I chose to set 32 filters for the two first conv2D layers and 64 filters for the two last ones. Each filter transforms a part of the image (defined by the kernel size) using the kernel filter. The kernel filter matrix is applied to the whole image. Filters can be seen as a transformation of the image.

The CNN can isolate features that are useful everywhere from these transformed images (feature maps).

The second important layer in CNN is the pooling (MaxPool2D) layer. This layer simply acts as a down-sampling filter. It looks at the 2 neighboring pixels and picks the maximal value. These are used to reduce the computational cost, and to some extent also reduce overfitting. We have to choose the pooling size (i.e. the area size pooled each time) more the pooling dimension is high, the more the downsampling is important.

Combining convolutional and pooling layers, CNN is able to combine local features and learn more global features of the image.

Dropout is a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their weights to zero) for each training sample. This drops randomly a proportion of the network and forces the network to learn features in a distributed way. This technique also improves generalization and reduces overfitting.

'relu' is the rectifier (activation function max(0,x). The rectifier activation function is used to add nonlinearity to the network.

The Flatten layer is used to convert the final feature maps into a single 1D vector. This flattening step is needed so that you can make use of fully connected layers after some convolutional/max pool layers. It combines all the found local features of the previous convolutional layers.

In the end, I used the features in two fully connected (Dense) layers which is just artificial an neural networks (ANN) classifier. In the last layer(Dense(10,activation="softmax")) the net outputs distribution of the probability of each class.

Link to the code at kaagle

https://www.kaggle.com/code/sid321axn/step-wise-approach-cnn-model-77-0344-accuracy