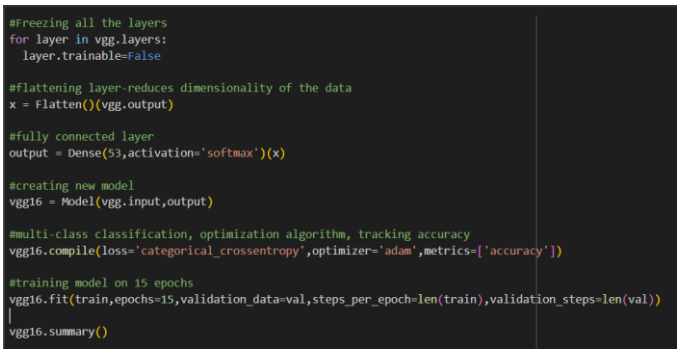
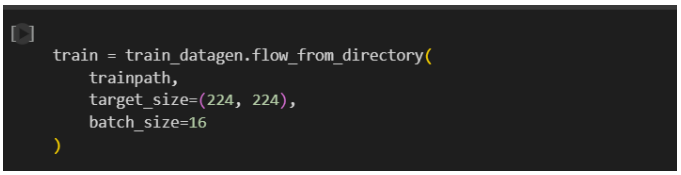
Model Optimization and Tuning Phase

Date	19 June 2025
Team ID	SWTID1749908722
Project Title	CardMaster: Intelligent Playing Card Recognition using Transfer Learning

Model Optimization and Tuning Phase

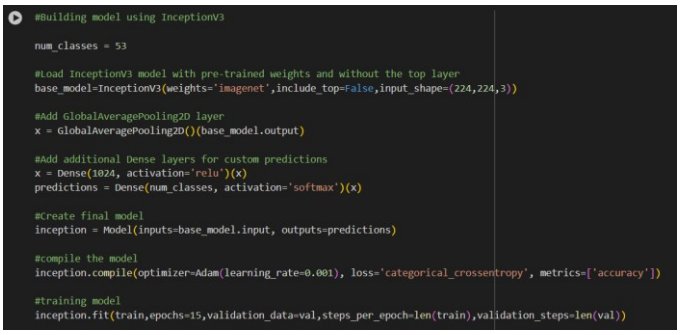
The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters	Performance metrics
VGG16	Code Screenshot:   optimizer='adam' : Adaptive optimizer, speeds up convergence loss='categorical_crossentropy' : Best for multi-class classification	Training Accuracy: 0.9134 Training Loss: 0.430 Validation Accuracy: 0.8377 Validation Loss: 1.2883 Observation: Overfitting – training_acc > val_accuracy by a wide margin.

	metrics=['accuracy'] : Tracks accuracy during training epochs=15 : Number of training iterations batch_size=16 : Images processed per training step target_size=(224, 224) : Image resized for model input trainable=False : Freezes pretrained VGG layers	

Xception	Code Screenshot:
	
	`#Building using Xception Transfer Learning Model`
	`num_classes = 53`
	
	`#Load Xception model with pre-trained weights and without the top layer`
	`xception = Xception(weights='imagenet', include_top=False, input_shape=(224,224,3))`
	
	`#Add GlobalAveragePooling2D layer`
	`x = GlobalAveragePooling2D()(xception.output)`
	
	`#Add additional Dense layers for custom predictions`
	`x = Dense(1024, activation='relu')(x)`
	`predictions = Dense(num_classes, activation='softmax')(x)`
	
	`#Create final model`
	`xcep = Model(inputs=xception.input, outputs=predictions)`
	
	`#compile the model`
	`xcep.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])`
	
	`#training model`
	`xcep.fit(train,epochs=15,validation_data=val,steps_per_epoch=len(train),validation_steps=len(val))`
	
	optimizer=Adam(learning_rate=0.001) : Adam with custom LR
	
	loss='categorical_crossentropy' : For multi-class output
	
	epochs=15
	
	batch_size=16
	
	Dense(1024, activation='relu') : Added custom dense layer
	
	target_size=(224, 224)
	
	include_top=False : Removes original Fully Connected layers of base model
		Training Accuracy: 0.9160
		
		Training Loss: 0.2870
		
		Validation Accuracy: 0.9427
		
		Validation Loss: 0.1979
		
		Observation:
		
		Best overall - excellent generalization, no signs of overfitting or underfitting.

Inception V3	Code Screenshot:  optimizer=Adam(learning_rate=0.001) : Adam with custom LR loss='categorical_crossentropy' : For multi-class output epochs=15 batch_size=16 Dense(1024, activation='relu') : Added custom dense layer target_size=(224, 224) include_top=False : Removes original Fully Connected layers of base model	Training Accuracy: 0.8072 Training Loss: 0.6436 Validation Accuracy: 0.9132 Validation Loss: 0.3975 Observation: Possible underfitting - value_acc > train_acc, model might not have learned enough.

Final Model Selection Justification (2 Marks):

Final Model	Reasoning

Model : Xception	Xception was chosen as the final optimized model based on its strong performance and computational efficiency.
	• Achieved the **highest validation accuracy** of 94.27% and the **lowest validation loss** of 0.1979
	• Training accuracy of 91.60% with a low loss of 0.2870, showing **stable learning with no overfitting**
	• Outperformed VGG16 and InceptionV3 in **generalization and consistency**
	• Uses **depthwise separable convolutions**, reducing parameter count and improving efficiency
	• Offers a good **balance of accuracy, training time, and resource usage**
	• Suitable for deployment due to **lower computational and time complexity**
	`1/1 ——— 0s 40ms/step`
	`Predicted Class Index: 31`
	`Predicted Class Name: queen of hearts`
	• Took lowest time out of all three models for card prediction.