

Using TinyML to Classify Playing Cards Types

Israel Rodrigues Dutra

São Paulo
2024

Objective

Make an artificial intelligence model to classify four types of playing cards and use edge devices to run the model.

Materials

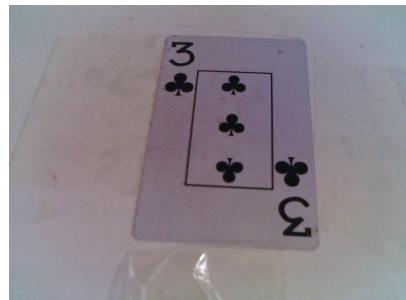
- ESP 32 CAM micro controller;
- Smartphone;
- Playing cards deck;
- Edge Impulse platform;
- Google Colab platform;
- Arduino IDE platform.

Methods

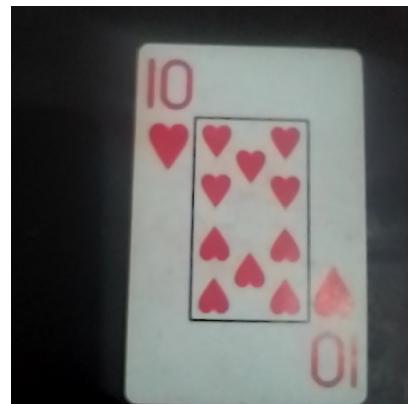
First, it was needed to create a dataset. Using ESP32 CAM to take images and store them in a SD card, it was possible to create folders with images of cards from A-K going through all types. Photos were also taken with a smartphone to add more data to the model. With a created dataset, it was uploaded to the edge impulse platform to create a machine learning model and finally the model was deployed on the smartphone and the ESP32 CAM.

Results

ESP32 CAM dataset image example



Smartphone dataset image example



Edge Impulse Model

Data acquisition

SAMPLE NAME	LABEL	ADDED
hearts.4l3nql0v	Hearts	Yesterday, 1...
spades.4l3n7i81	Spades	Yesterday, 1...
hearts.4l3nr5h8	Hearts	Yesterday, 1...
Diamonds.4l57ef0d	Diamonds	Yesterday, 1...

Impulse design

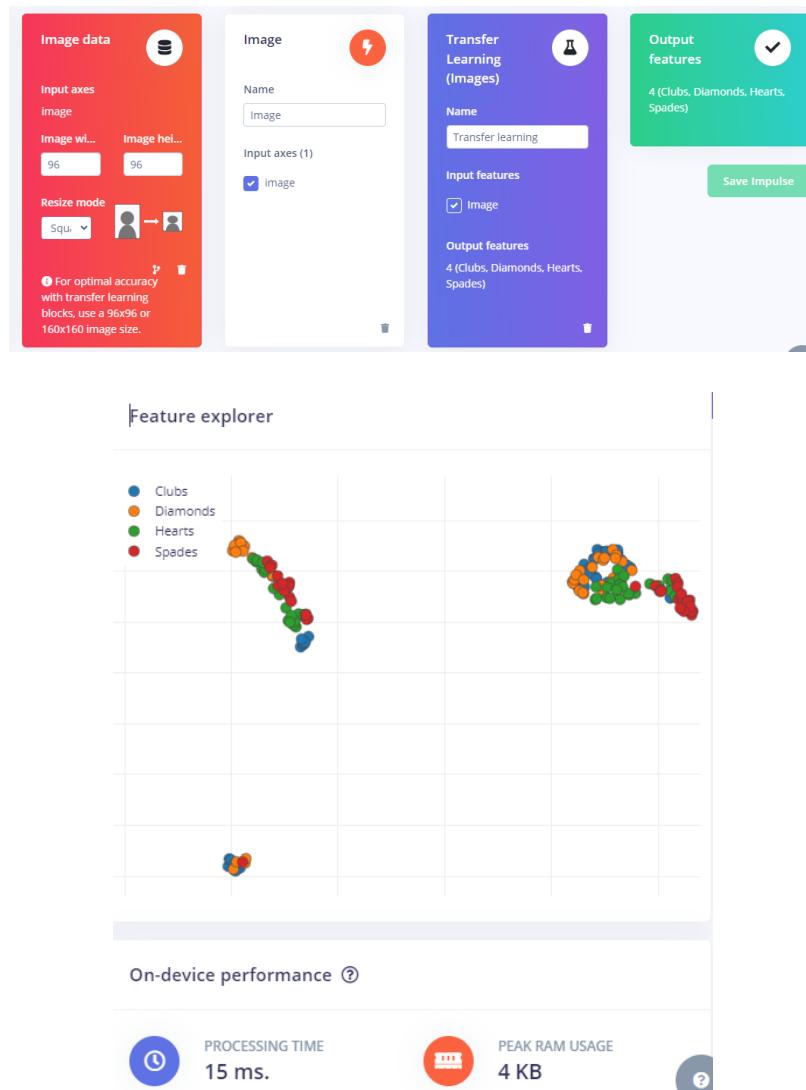


Figure 1: Target device ESP-EYE

Transfer Learning

Training settings

Number of training cycles [?](#)

Use learned optimizer [?](#)

Learning rate [?](#)

Data augmentation [?](#)

Advanced training settings

Neural network architecture

Input layer (9,216 features)



MobileNetV2 96x96 0.35 (final layer: 16 neurons, 0.1 dropout)

Choose a different model

Last training performance (validation set)



ACCURACY
88.6%

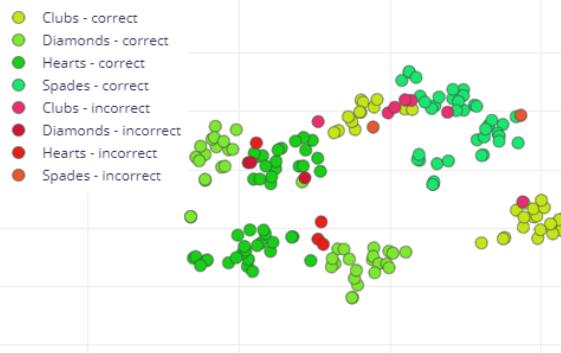


LOSS
0,31

Confusion matrix (validation set)

	CLUBS	DIAMONDS	HEARTS	SPADES
CLUBS	81.8%	0%	0%	18.2%
DIAMONDS	0%	85.7%	14.3%	0%
HEARTS	0%	14.3%	85.7%	0%
SPADES	0%	0%	0%	100%
F1 SCORE	0.90	0.86	0.86	0.91

Data explorer (full training set)



On-device performance



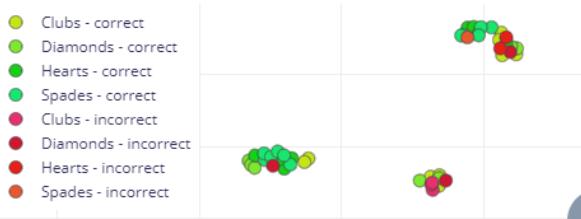
Model testing

Model testing results



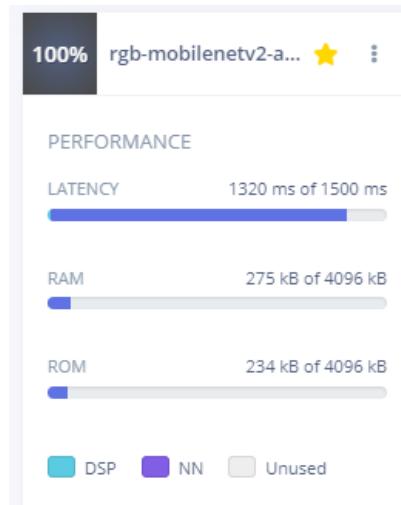
	CLUBS	DIAMONDS	HEARTS	SPADES	UNCERTAIN
CLUBS	75%	0%	16.7%	0%	8.3%
DIAMONDS	10%	70%	20%	0%	0%
HEARTS	0%	10%	80%	0%	10%
SPADES	0%	0%	0%	91.7%	8.3%
F1 SCORE	0.82	0.78	0.73	0.96	

Feature explorer



EON Tuner

EON Tuner is used to find the best model possible based on your target device.



Training settings

Number of training cycles: 20

Use learned optimizer:

Learning rate: 0.0005

Data augmentation:

Advanced training settings ▾

Neural network architecture

Input layer (27,648 features)

Model Model version: Quantized (int8)

Last training performance (validation set)

	ACCURACY 100.0%		LOSS 0,06
--	--------------------	--	--------------

Confusion matrix (validation set)

	CLUBS	DIAMONDS	HEARTS	SPADES
CLUBS	100%	0%	0%	0%
DIAMONDS	0%	100%	0%	0%
HEARTS	0%	0%	100%	0%
SPADES	0%	0%	0%	100%
F1 SCORE	1.00	1.00	1.00	1.00

Model testing results

ACCURACY 90.91%

	CLUBS	DIAMONDS	HEARTS	SPADES	UNCERTAIN
CLUBS	75%	0%	0%	8.3%	16.7%
DIAMONDS	0%	90%	0%	0%	10%
HEARTS	0%	0%	100%	0%	0%
SPADES	0%	0%	0%	100%	0%
F1 SCORE	0.86	0.95	1.00	0.96	

Live classification using smartphone

Classification result

Summary

Name	testing.4lfj8ppu
Label	testing

CATEGORY	COUNT
Clubs	1
Diamonds	0
Hearts	0
Spades	0
uncertain	0

RAW DATA
testing.4lfj8ppu

Raw features

0x191b1a, 0x191b1b, 0x1a1c1b, 0x1b1d1c, 0x1c1e1d, 0x

Image

- Clubs
- Diamonds
- Hearts
- Spades

Live classification from smartphone model

ESP32 classification



```
esp32_camera.ino ...  
1 /* Edge Impulse Arduino examples  
Output Serial Monitor X  
Message (Ent New Line ▾ 115200 baud ▾  
Predictions (DSP: 17 ms., Classification:  
    Clubs: 0.89062  
    Diamonds: 0.00391  
    Hearts: 0.00391  
    Spades: 0.09766  
Predictions (DSP: 17 ms., Classification:  
    Clubs: 0.70312  
    Diamonds: 0.00000  
    Hearts: 0.00000  
    Spades: 0.29688
```

