Guía de Rappy Fruits

Diego Iván Perea Montealegre (2185751) diego.perea@uao.edu.co

Esteban Ramos Gomez (2181034) esteban.ramos@uao.edu.co

Brayan Fernando Castro Balanta (2181032) brayan.castro@uao.edu.co

Facultad de Ingeniería, Universidad Autónoma de Occidente

Cali, Valle del Cauca

Esta guía contiene la información necesaria del funcionamiento de la aplicación “Rappy Fruits” y de cómo se debería de utilizar, esta utiliza CNN (red neuronal convolucional) que en el cual clasificará 5 clases de frutas como lo son : Bananas , manzanas , limones , Fresas y naranjas ;La aplicación estará conectada a través de bluetooth , en el que permitirá enviarle a una board (Arduino Uno) una acción , esta acción está en encender un led en la clase de fruta clasificada , dicho esto se mostrará primero la parte de la interfaz gráfica .

Figura 1 . Visualizacion del logo de la aplicaccion de “Rappy Fruits”

El almaenamiento de la aplicación es de 8.74MB lo que permite la facilidad del funcionamiento , a pesar de utilizar CNN esta responde bien en sistemas embebidos tan pequeño como lo es un celular.

La aplicación fue desarrolada a traves de la plataforma de desarrollo appinventor que cn la facilidad de programacion en bloque , hace que sea facil de utilizar y entendder para las demas personas , ademas de su clasificador propio de imágenes que fue tilizado para construir el modelo .En este se utilizo data sets de paginas como kaggle y mendeley en el cual el aporte fue de las repetivas frutas a clasificar : Bananas, manzanas, limones, Fresas y naranjas

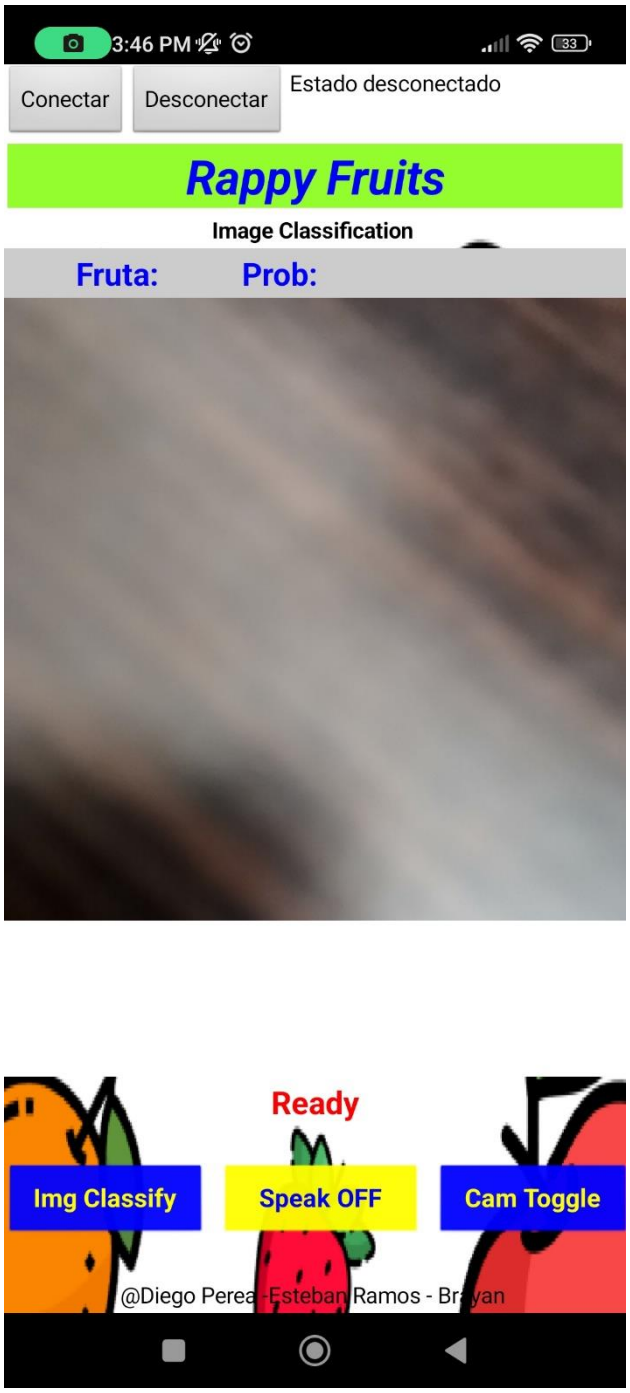


Figura 2 . Visualizacion de Interfaz de “Rappy Fruits”

Cada funcionamiento de la aplicación estará efectuada con botones que realizaran la acción seleccionada , en la aplicación como anteriormente se detalló se clasifica cinco clases de frutas en la que con el botón de “img classify ” clasificara la imagen que la cámara este visualizando en el momento , el botón “Cam toggle” cambiará la cámara a frontal , el botón “Speak” dará voz a la fruta clasificada y su probabilidad , esta función fue tomada en cuenta por las personas que no puedan tener una visión reducida y así puedan conocer que fruta fue clasificada; el botón “Conectar” se conecta a bluetooth y con el botón “Desconectar” se desconecta del bluetooth.

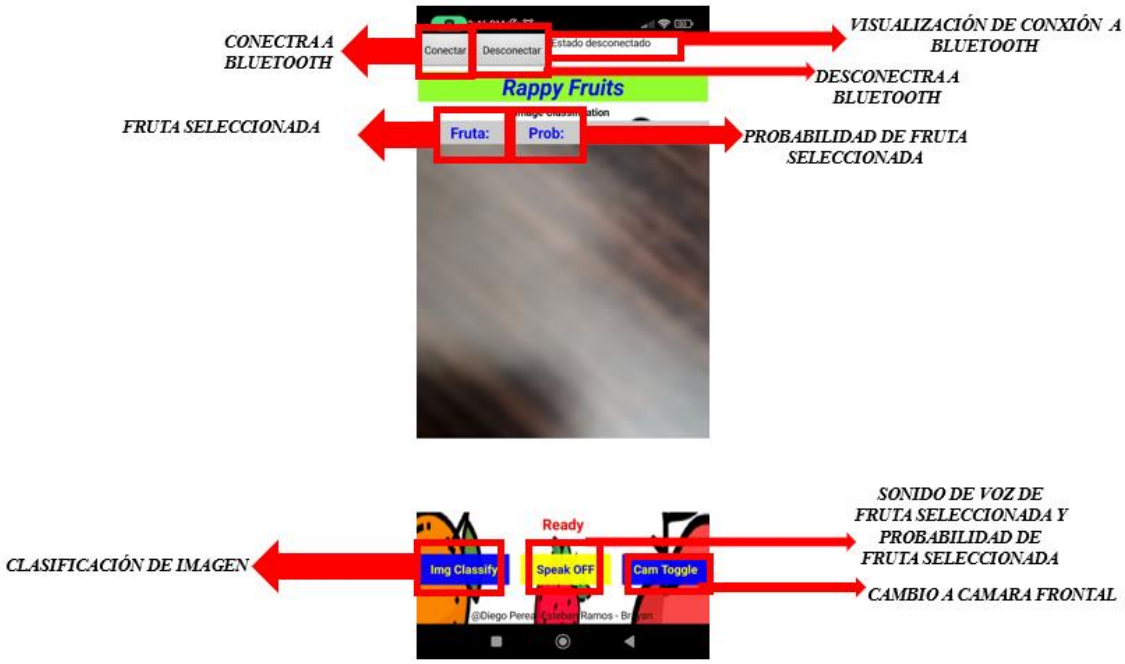


Figura 3 . Funcionamiento de botones de “Rappy Fruits”

A continuacion se muestra el codigo de desarrollo del funcionamiento de “Rappy Fruits” , para conocer las acciones que realiza en cada proceso.

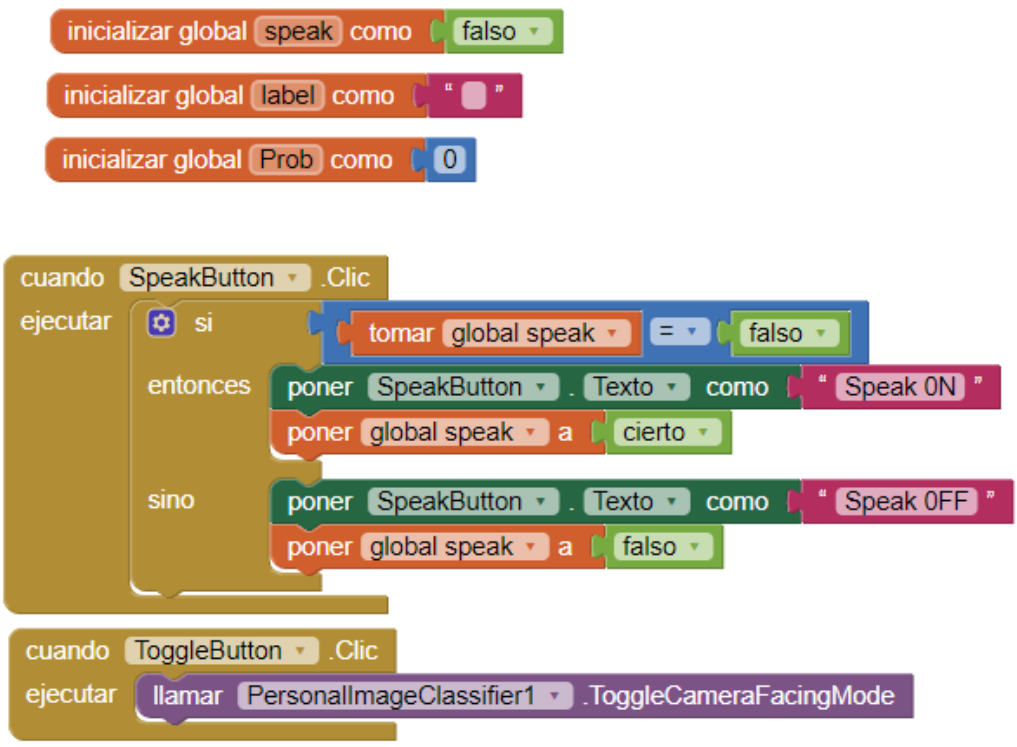


Figura 4 . Codigo de iniciacion y ejecucion de speak y toggle de “Rappy Fruits”

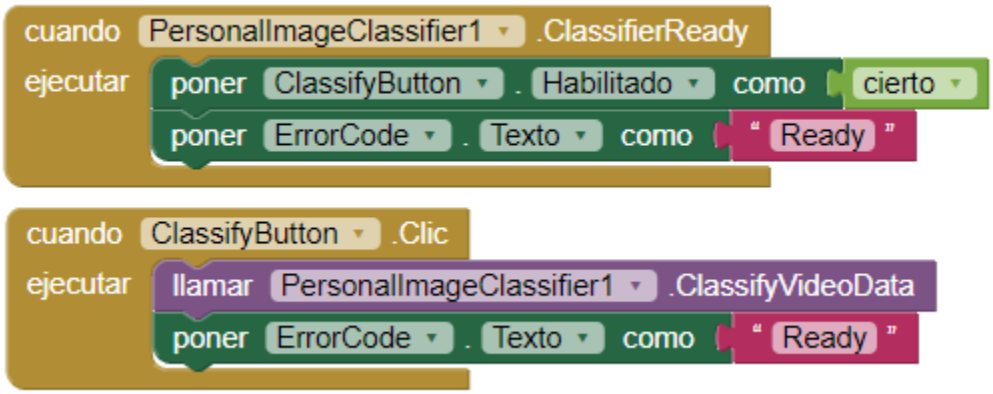


Figura 5 . Llamada a prepar el modelo de clasificacion de “Rappy Fruits”

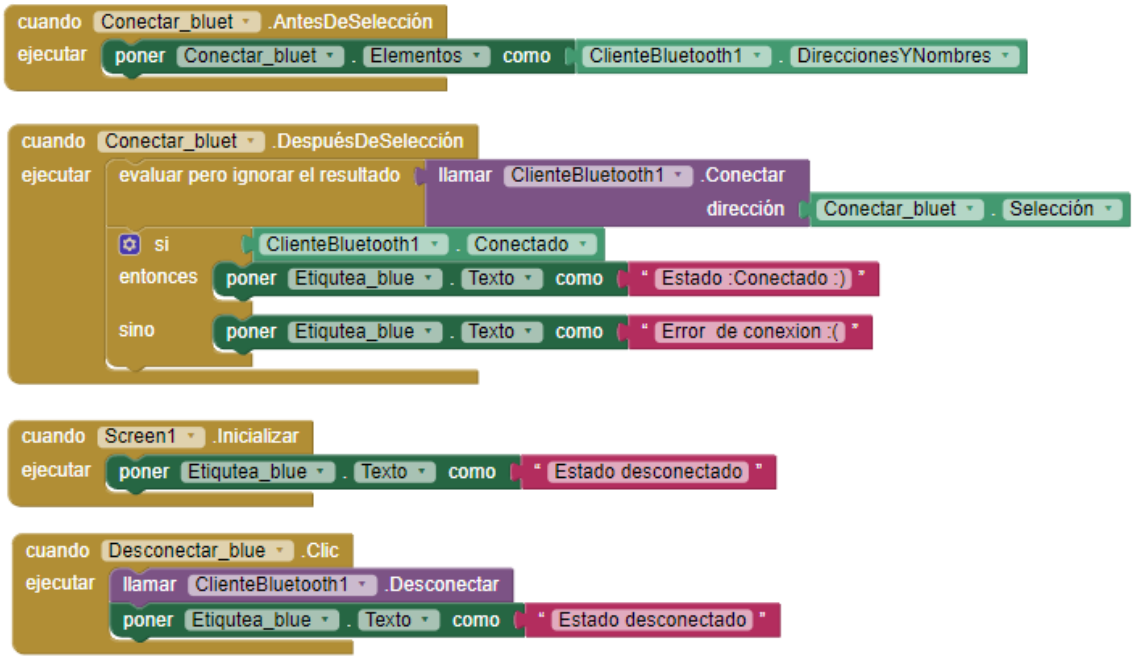


Figura 6 . Conexión y funcionamiento de bluetooth de “Rappy Fruits”

Figura 7 . Codigo cuando haya error en clasificacion de “Rappy Fruits”

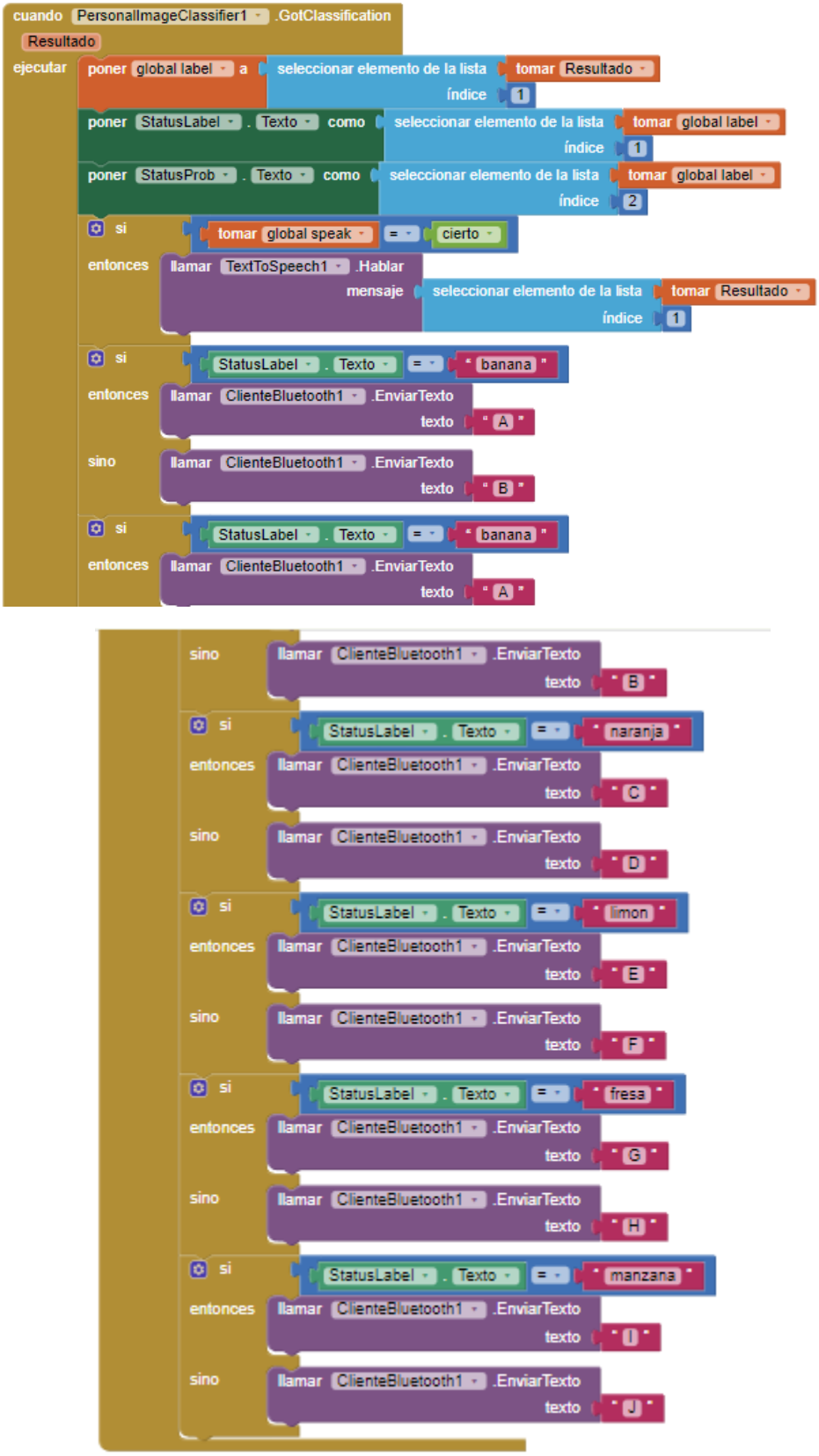


Figura 8 . Metodo de clasificacion de “Rappy Fruits”

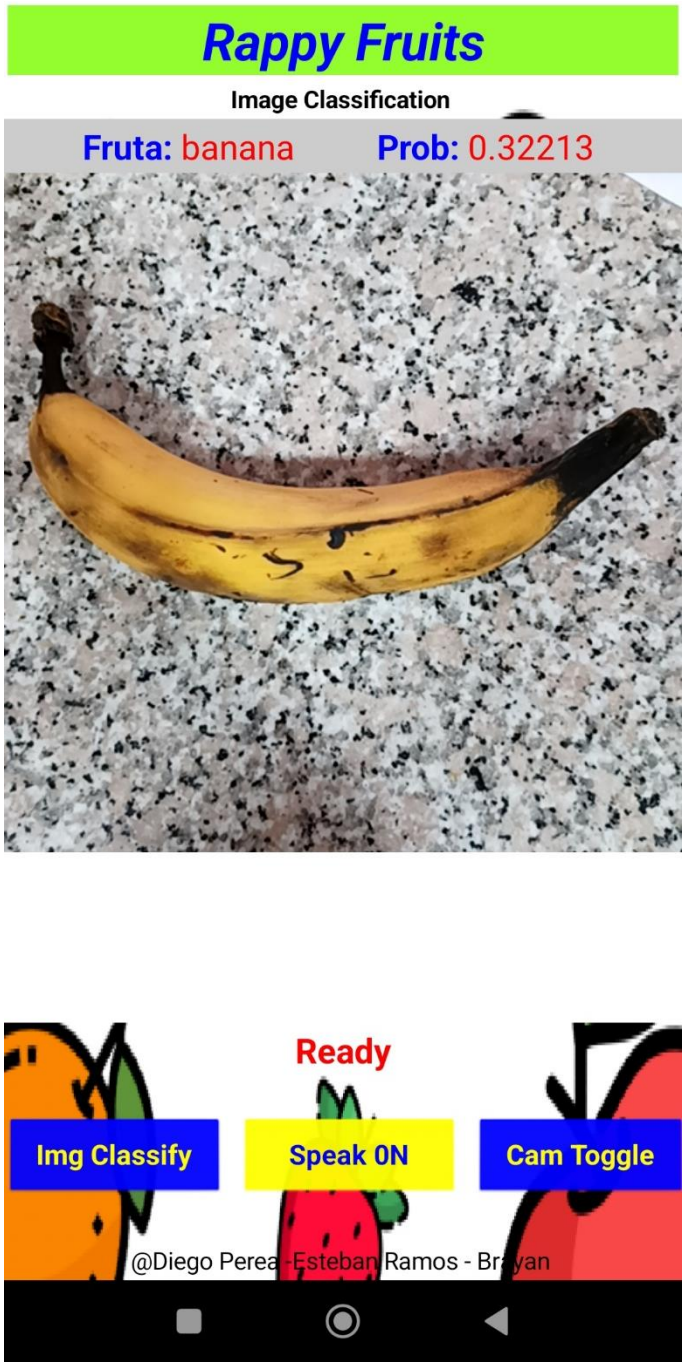


Figura 9 .Fruta clasificada de “Rappy Fruits”

Hay que aclarar que el modelo se contruyo gracias a la plataforma de “https://classifier.appinventor.mit.edu” que en el cual se pudo darle a cada clase su respetiva etiqueta de Banana, manzana, limón, Fresa y naranja, en el que se utilizó una red neuronal de tipo convolucional.

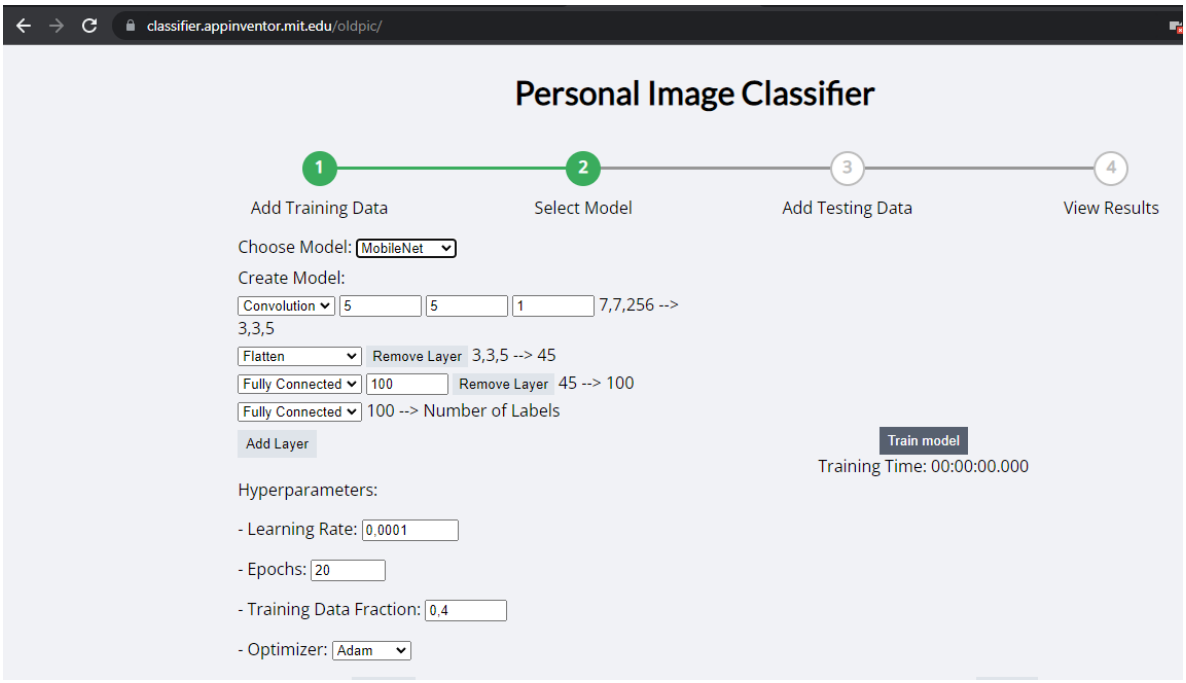


Figura 10 . Parametros de clasificacion de “Rappy Fruits”

Los parametros en la construccion del modelo “Rappy Fruits” se tomaron los recomendados como se observa en la anterior imagen , solos e modifico el numero de epocas a 40 , dado que se habia intentado con 20 epocas peo no era tan eficaz ,incluso se intento modelar con mas de 200 imágenes en cada una de las clases pero esto la plataforma no lo aceptaba debido a que se quedaba cargando intermitentemente , por lo que se tomo 40 imágenes para cada clase de frutas en el entrenamiento del modelo ; Este dataset fue tomado en combinacion de dos data set en el que fueron las bananas , naranjas y manzanas de kaggle[14] y limon , fresa de medeley [13].

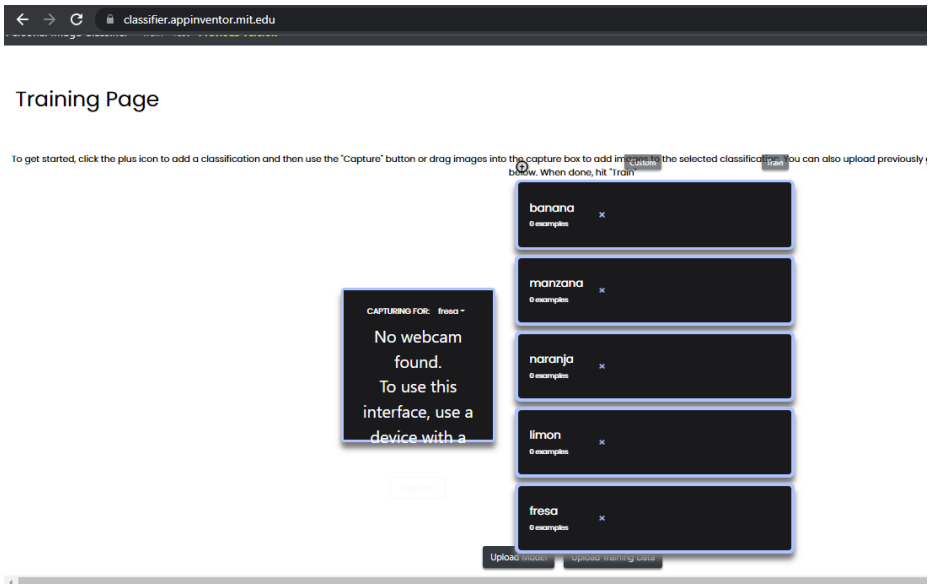


Figura 11 . Labels de clasificacion de “Rappy Fruits”

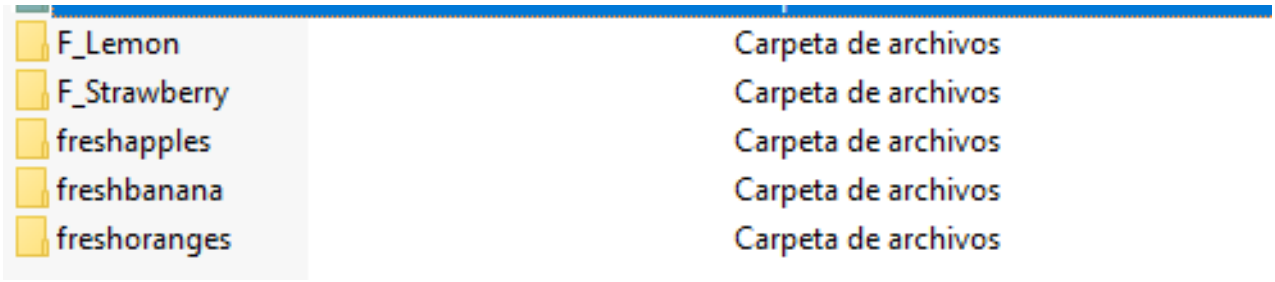


Figura 11 . Data set clasificacion de “Rappy Fruits”

En cada clase de las frutas se sleeciono las 40 primeras imágenes para entrenarlas y asi generar el model con el nombre de “model.mdl” ya teniendo el modelo se importa a la aplicación de desarrollo en appinventor y comienza la programacion en bloques observada en las figuras 3 hasta la figura 8.

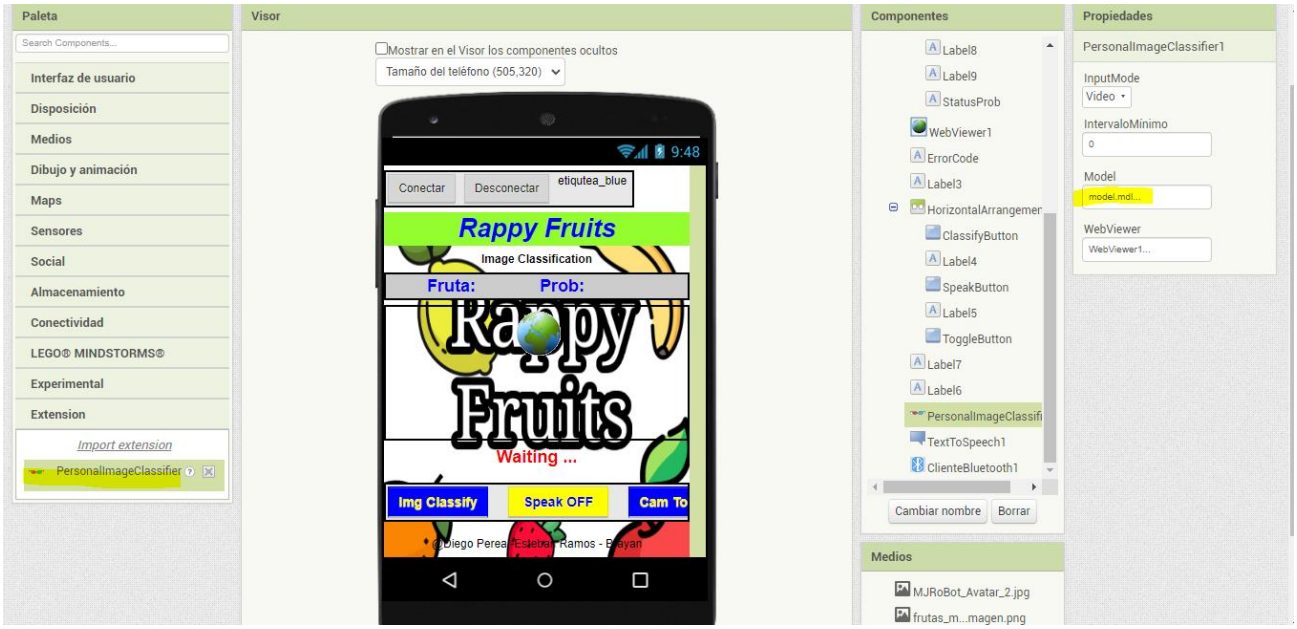


Figura 12 . Insercion del modelo.mdl de “Rappy Fruits”

Pero al observar la aplcacion una mejora de esta es que la probabilidad de la fruta sea mayor de 0.6 , para asi tener la certeza de lo que se esta clasificando se la fruta indicada, se puede modificar la figura 8 , de la siguiente manera .

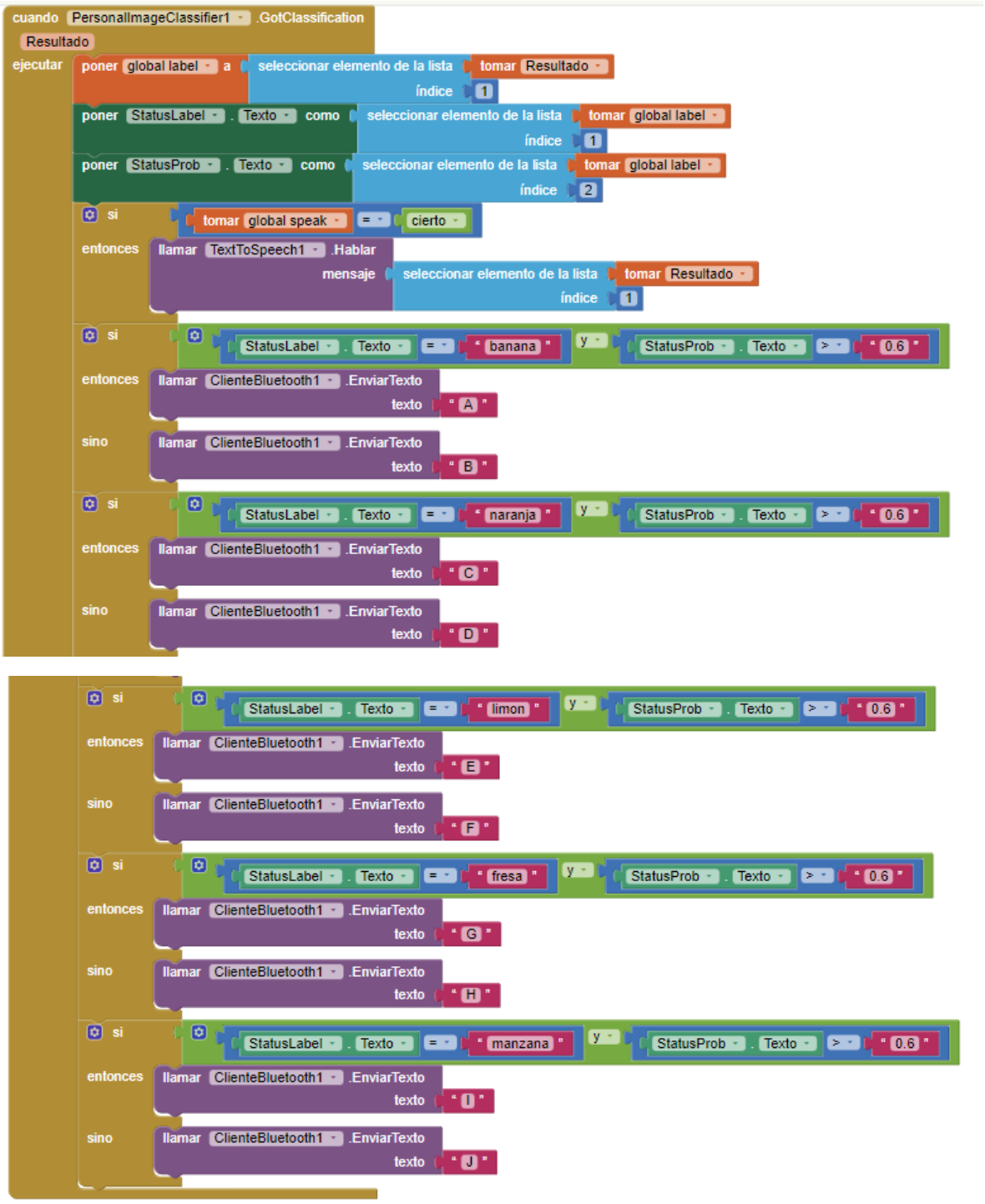


Figura 13 Modificacion para aumentar la precision de clasificacion de “Rappy Fruits”

El funcionamiento del arduino es cuando reciba datos del bluetooth este activa ON en el led seleecionado de la fruta clasificada

```ino
#include <Arduino.h>

//Variables asociadas a los dos LEDs que se van a controlar
int led_1 = 2;
int led_2 = 3;
int led_3 = 4;
int led_4 = 5;
int led_5 = 6;
char valor;  //Variable para indicar que llega una orden

void setup() {

  pinMode(led_1, OUTPUT);
  pinMode(led_2, OUTPUT);
  pinMode(led_3, OUTPUT);
  pinMode(led_4, OUTPUT);
  pinMode(led_5, OUTPUT);
  Serial.begin(9600);
}

void loop() {

  if (Serial.available())  //Si el puerto serie (Bluetooth) está disponible
  {
    valor = Serial.read();  //Lee el dato entrante via Bluetooth
    //Banana
    if (valor == 'A')   //Si el dato que llega es una A
    {
      digitalWrite(led_1, HIGH);  //Enciende el LED 1
    }
    if (valor == 'B')   //Si el dato que llega es una B
    {
      digitalWrite(led_1, LOW);   //Apaga el LED 1
    }
    //Naranja
    if (valor == 'C')   //Si el dato que llega es una B
    {
      digitalWrite(led_2, HIGH);   //Apaga el LED 1
    }
    if (valor == 'D')   //Si el dato que llega es una B
    {
      digitalWrite(led_2, LOW);   //Apaga el LED 1
    }
    //Limon
    if (valor == 'E')   //Si el dato que llega es una B
    {
      digitalWrite(led_3, HIGH);   //Apaga el LED 1
    }
    if (valor == 'F')   //Si el dato que llega es una B
    {
      digitalWrite(led_3, LOW);   //Apaga el LED 1
    }
    //Fresa
    if (valor == 'G')   //Si el dato que llega es una B
    {
      digitalWrite(led_4, HIGH);   //Apaga el LED 1
    }
    if (valor == 'H')   //Si el dato que llega es una B
    {
      digitalWrite(led_4, LOW);   //Apaga el LED 1
    }
    //Manzana
    if (valor == 'I')   //Si el dato que llega es una B
    {
      digitalWrite(led_5, HIGH);   //Apaga el LED 1
    }
    if (valor == 'J')   //Si el dato que llega es una B
    {
      digitalWrite(led_5, LOW);   //Apaga el LED 1
    }

  }
}

```

Figura 14 . Codigo de arduino de “Rappy Fruits”

A continuacion se muestra evidencias fisicas de las conexiones pertinentes de la aplicación , con el bluetooth y la board (Arduino uno), En la figura 15 , se observa que cada led en cada frruta representa la clasifiacion que se realice en la aplicación “Rappy Fruits” , por ejemplo si en la aplicación clasifica la imagen como una “banana” (Figura 9) el led situado en la banana prenderá y asi sucesivamente frente a las demas clases dee frutas.Cabe resaltar que cuando se suba el codigo de arduino se debe desconectar el bluetooth debido a que causas errores al moente de subir el codigo .

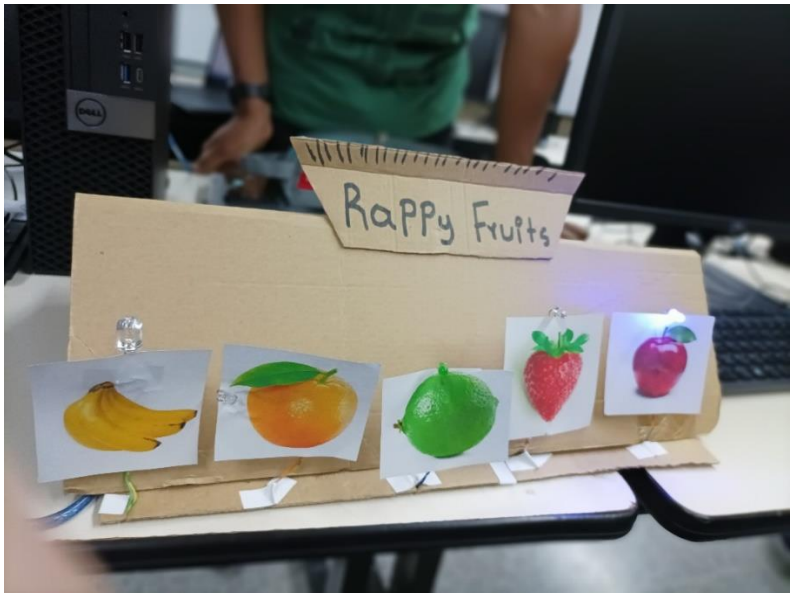


Figura 15 . Montaje de conexiones y una estetica del trabajo realizado.

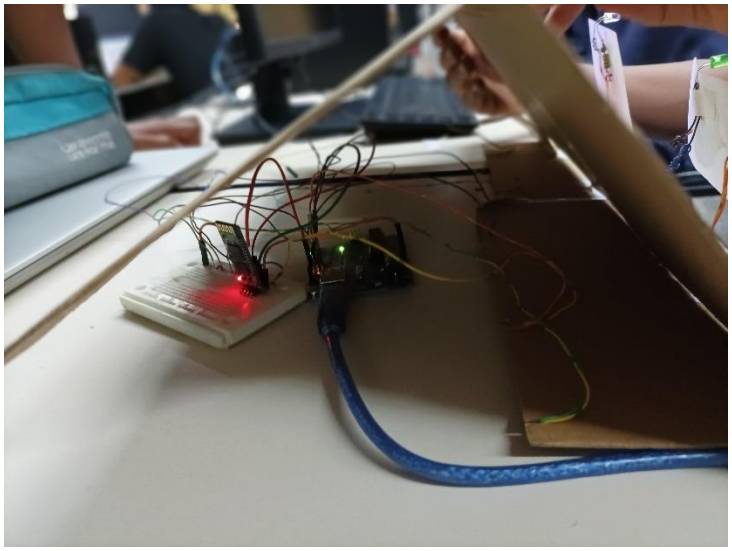
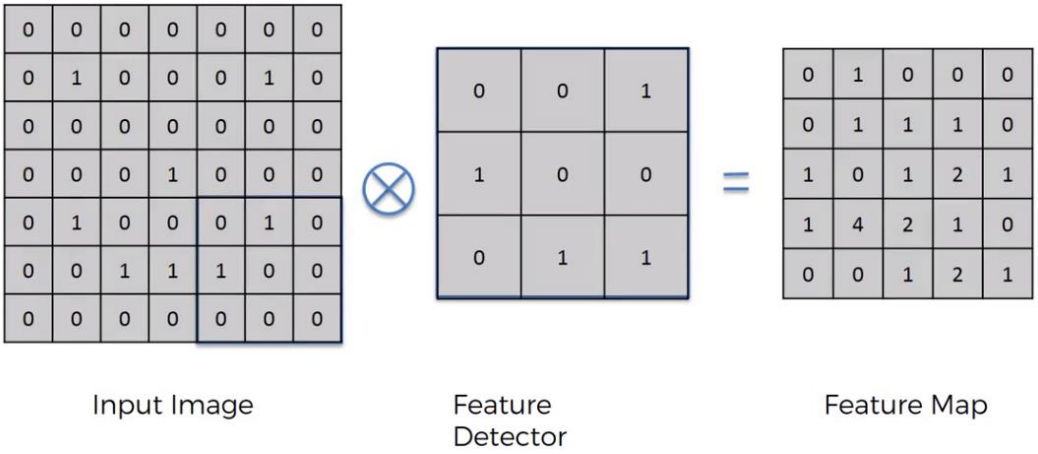


Figura 16 . Conexiones con bluettoth y board (Arduino Uno)

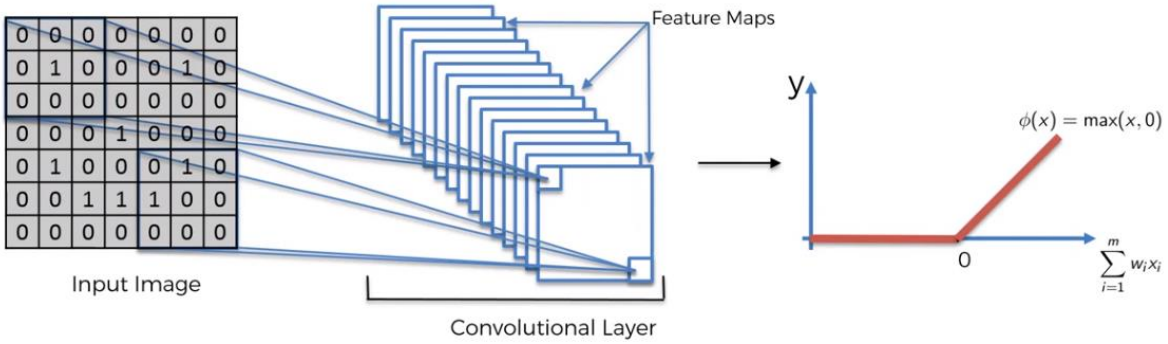
La red neuronal convolucional CNN es un aprendizaje profundo supervisado utilizado para la visión por computadora . El proceso de Redes Neuronales Convolucionales se puede dividir en cinco pasos:

Convolución , Max Pooling , Flattening , Full Connection . PASO 1-Convolución En la base de Convolución hay un filtro también llamado Detector de Características o Kernel . Básicamente, multiplicamos la parte de la imagen por el filtro y verificamos la coincidencia de cuántos 1 tienen en común.



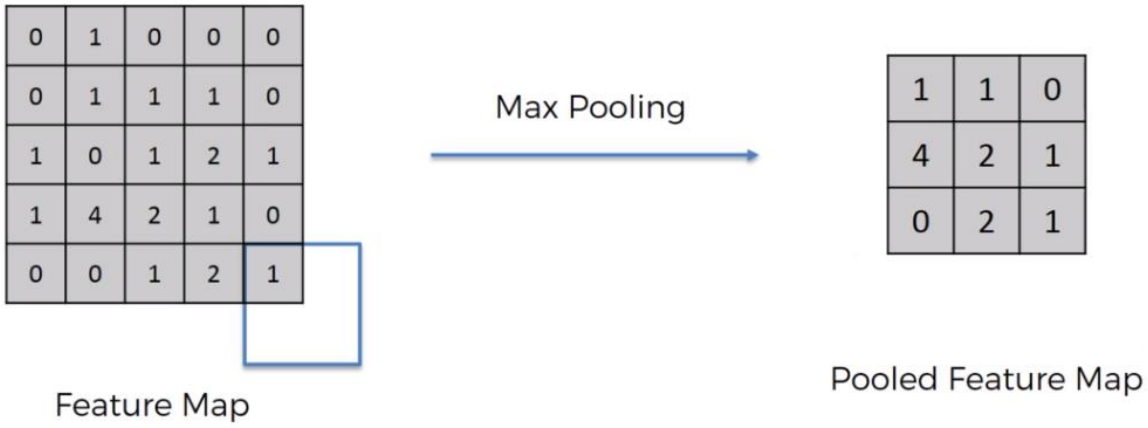
La imagen resultante en la parte superior derecha de la figura anterior se llama Feature Map . Esa es esencialmente la cantidad de características convolucionadas. El resultado es una reducción de la información de la imagen de entrada. Cuando tenemos un 2 significa que reducimos la imagen, pero un 4 significa que reducimos aún más la imagen original, y eso hace que la imagen sea más fácil de procesar. La pregunta es si hay pérdida de información aplicando el filtro Feature Detector. Cuanto mayor sea el número que tenemos en el mapa de funciones, mejor es el proceso de filtrado y eso significa que no estamos perdiendo muchas funciones. Esencialmente, creamos múltiples mapas de características como filtros de detección de características que necesitamos (por ejemplo, detección de bordes, detección de desenfoque, detección de relieve).

PASO 2 - Capa ReLU Esta es la función de Unidad lineal rectificada , un paso adicional además de Convolución. La razón por la que se utiliza ReLU es para aumentar la no linealidad. La razón por la que queremos aumentar la no linealidad es porque las imágenes son altamente no lineales, por eso queremos romper la linealidad.



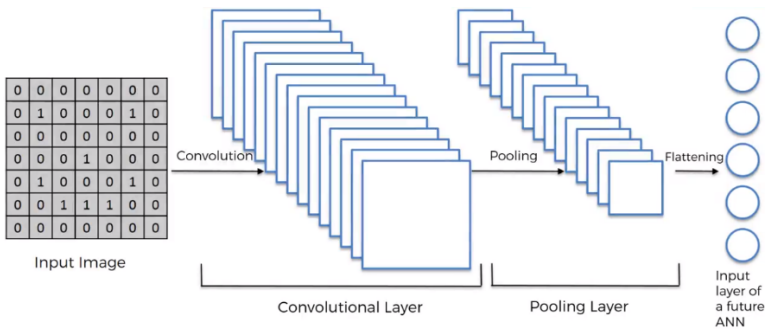
Lo que hace ReLU, por ejemplo, en una imagen en blanco y negro, es eliminar el componente lineal creado por las sombras en una imagen. De hecho, las sombras se muestran en una imagen como una progresión lineal de escala de grises, y podemos excluir esta información demasiado detallada usando ReLU.

PASO 3 - Max Pooling Por lo general, podemos tener una característica (por ejemplo, un perro) de una imagen presentada en muchas posiciones u orientaciones diferentes. Tenemos que estar seguros de que nuestra red neuronal tiene una propiedad llamada Invariancia espacial . Esto significa que no importa dónde se ubican las funciones en diferentes entornos, o si están más cerca o un poco más separadas: nuestra red neuronal debe tener la flexibilidad para encontrar la función. Para conseguir eso usamos Pooling .

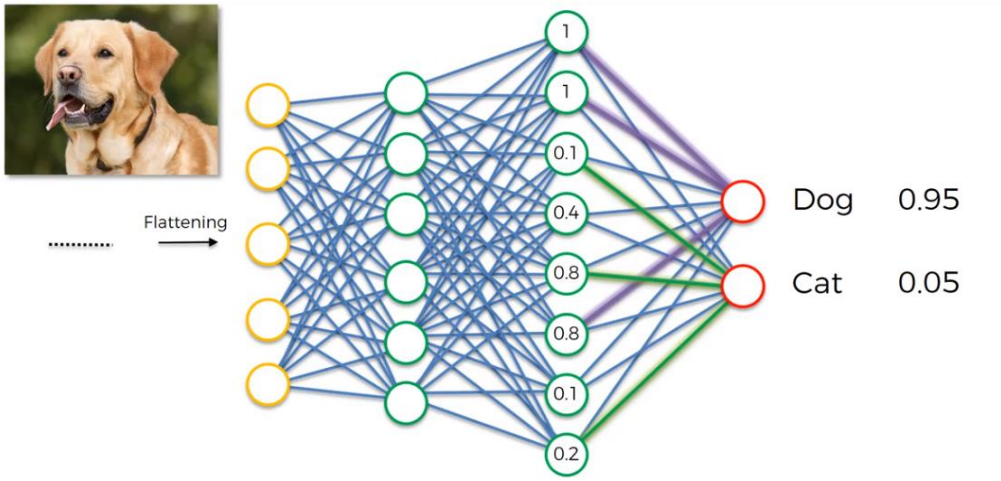


De la figura anterior, usamos Max Pooling . Tomamos un cuadro de 2x2 píxeles del Feature Map creado previamente, encontramos el valor máximo y lo reportamos en el Pooled Feature Map . Repetimos este proceso moviendo el cuadro 2x2 en el Feature Map. El resultado es que todavía estamos preservando las características, y debido a que estamos tomando el máximo de píxeles, estamos teniendo en cuenta cualquier distorsión y posible distorsión espacial o de textura o de otro tipo. Además, otro beneficio de la agrupación es que estamos reduciendo la cantidad de parámetros y, por lo tanto, estamos evitando el sobreajuste. eso es como en los humanos, porque lo importante para la visión no es ver exactamente todas las características que pueden ser ruidosas, sino que los humanos desprecien la información innecesaria. Hay muchos tipos de agrupación, y evaluar el tipo de agrupación es crucial. Hay un artículo muy importante sobre eso que se llama: Evaluación de las operaciones de Pooling en Arquitecturas Convolucionales para el Reconocimiento de Objetos por Dominik Scherer . También hay una herramienta en línea muy útil que podemos usar para tener una mejor idea de lo que sucede durante el proceso de convolución y agrupación, y podemos verificar la herramienta haciendo clic aquí .

PASO 4 - Aplanamiento El proceso de Aplanamiento simplemente significa reordenar el Mapa de características agrupadas en una sola columna. Hacemos eso, porque este vector ahora está listo para usarse como entrada de una ANN para su posterior procesamiento. Todo el proceso descrito hasta ahora se puede resumir en la siguiente figura.

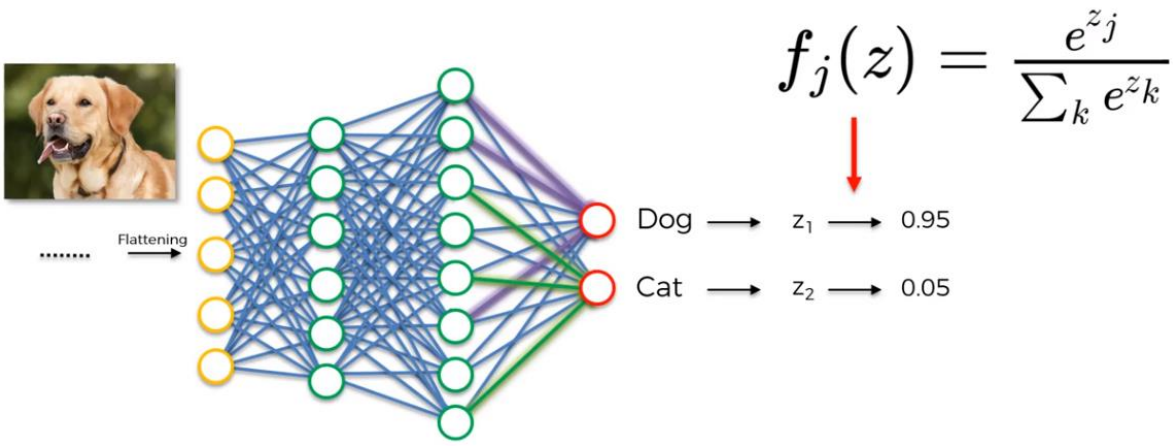


PASO 5 - Conexión completa El vector aplanado que describimos anteriormente, ahora se usa como entrada en una ANN completamente conectada . Con totalmente conectado queremos decir que la capa oculta está totalmente conectada. Esto es por definición una CNN. El propósito de esto es combinar nuestras funciones en más atributos para predecir las clases aún mejor. De hecho, la combinación de más atributos (por ejemplo, detección de bordes, detección de desenfoque, detección de relieve) ayuda a predecir mejor las imágenes. La siguiente figura muestra como ejemplo las neuronas activadas para un Perro



Los vínculos entre las neuronas y la salida de color violeta para el perro y verde para el gato nos indican cuáles son las neuronas importantes para el perro y el gato, respectivamente. La siguiente figura resume todos los pasos considerados hasta ahora.

Softmax y Cross Entropy; De la figura sobre la detección del perro, encontramos 0.95 Perro y 0.05 Gato. La pregunta es cómo estos dos valores suman 1. Eso es posible solo introduciendo la Función Softmax , la fórmula es la siguiente:



Como se describe en la fórmula de la figura anterior, la función Softmax es una generalización de la función logística y garantiza que nuestra predicción sume 1. La mayoría de las veces, la función Softmax está relacionada con la función de entropía cruzada . En CNN, luego de la aplicación de la Función Softmax, se trata de probar la confiabilidad del modelo utilizando como Función de Pérdida la Función de Entropía Cruzada , con el fin de maximizar el rendimiento de nuestra red neuronal. Hay

varias ventajas en el uso de la función de entropía cruzada. Una de las mejores es que si, por ejemplo, al comienzo de la retropropagaciónel valor de salida es mucho más pequeño que el valor real, el descenso del gradiente será muy lento. Debido a que Cross Entropy usa el logaritmo, ayuda a la red a evaluar incluso errores grandes.

Lóbulo occipital y CNN ,Es común vincular el lóbulo occipital del cerebro humano con CNN. De hecho, las CNN son las responsables de la visión artificial, el reconocimiento de imágenes y objetos, lo que las convierte en un vínculo perfecto con el lóbulo occipital.

Referencias

[1]"CNN and Softmax - Andrea Perlato", Andreaperlato.com, 2022. [Online]. Available: https://www.andreaperlato.com/aipost/cnn-and-softmax/. [Accessed: 10- Sep- 2022]

[2]"Personal Image Classifier", Classifier.appinventor.mit.edu, 2022. [Online]. Available: https://classifier.appinventor.mit.edu/. [Accessed: 10- Sep- 2022]

[3]"Personal Image Classifier: PICaboo", Appinventor.mit.edu, 2022. [Online]. Available: https://appinventor.mit.edu/explore/resources/ai/picaboo. [Accessed: 10- Sep- 2022]

[4]"Personal Audio Classifier", Appinventor.mit.edu, 2022. [Online]. Available: https://appinventor.mit.edu/explore/resources/ai/personal-audio-classifier. [Accessed: 10- Sep- 2022]

[5]"FINAL IJCAI_EDUAI_19__Personal_Image_Classifier.pdf", Google Docs, 2022. [Online]. Available: https://drive.google.com/file/d/18x1pGEoKrq_4ShkFdH2hYjoTtc0qyECa/view. [Accessed: 10- Sep- 2022]

[6]"Danny_Tang_Thesis.pdf", Google Docs, 2022. [Online]. Available: https://drive.google.com/file/d/1lfOVkajhYZaFtVGZnNWjQn6nBrBLWzE1/view. [Accessed: 10- Sep- 2022]

[7]Appinventor.mit.edu, 2022. [Online]. Available: https://appinventor.mit.edu/assets/files/Nikhil_Bhatia_MEng_Thesis.pdf. [Accessed: 10- Sep- 2022]

[8]Appinventor.mit.edu, 2022. [Online]. Available: https://appinventor.mit.edu/assets/files/CTE2020_paper_38.pdf. [Accessed: 10- Sep- 2022]

[9]M. Rovai) and A. Veggies, "App Inventor: EdgeML Image Classification: Fruits vs Veggies", Hackster.io, 2022. [Online]. Available: https://www.hackster.io/mjrobot/app-inventor-edgeml-image-classification-fruits-vs-veggies-b671da. [Accessed: 10- Sep- 2022]

[10]"GitHub - Mjrovai/APP_Inventor-ML_Projects: Machine Learning Projects with App Inventor", GitHub, 2022. [Online]. Available: https://github.com/Mjrovai/APP_Inventor-ML_Projects. [Accessed: 10- Sep- 2022]

[11]"APP_Inventor-ML_Projects/code at main · Mjrovai/APP_Inventor-ML_Projects", GitHub, 2022. [Online]. Available: https://github.com/Mjrovai/APP_Inventor-ML_Projects/tree/main/code. [Accessed: 10- Sep- 2022]

[12]Youtube.com, 2022. [Online]. Available: https://www.youtube.com/watch?v=OO7vKKuJ9a0. [Accessed: 10- Sep- 2022]

[13]C. Pachon Suescun, J. Pinzón Arenas and R. Jiménez-Moreno, "Spoiled and fresh fruit inspection dataset", Mendeley Data, 2022. [Online]. Available: https://data.mendeley.com/datasets/6ps7gtp2wg/1. [Accessed: 12- Sep- 2022]

[14]"Fruits fresh and rotten for classification", Kaggle.com, 2022. [Online]. Available: https://www.kaggle.com/datasets/sriramr/fruits-fresh-and-rotten-for-classification. [Accessed: 12- Sep- 2022]