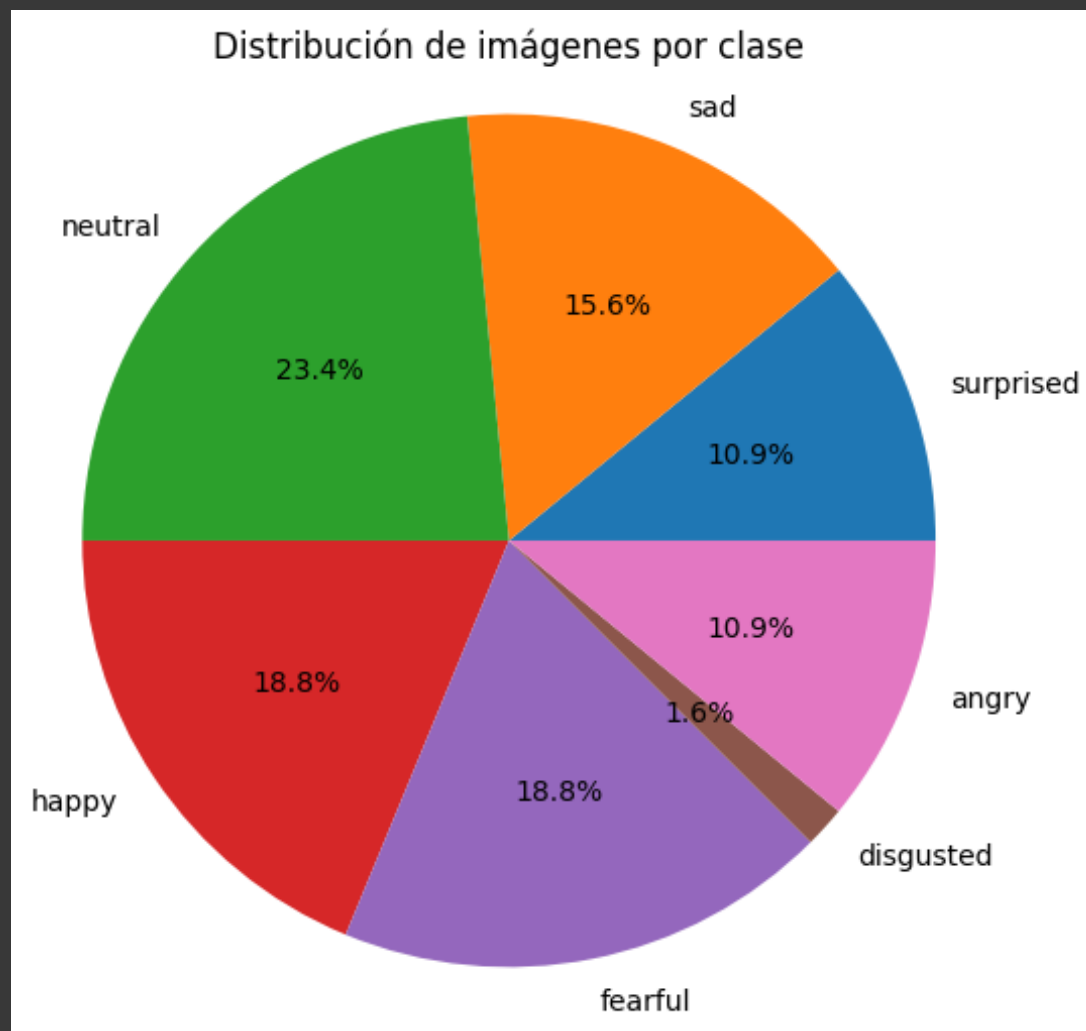


Facial expression

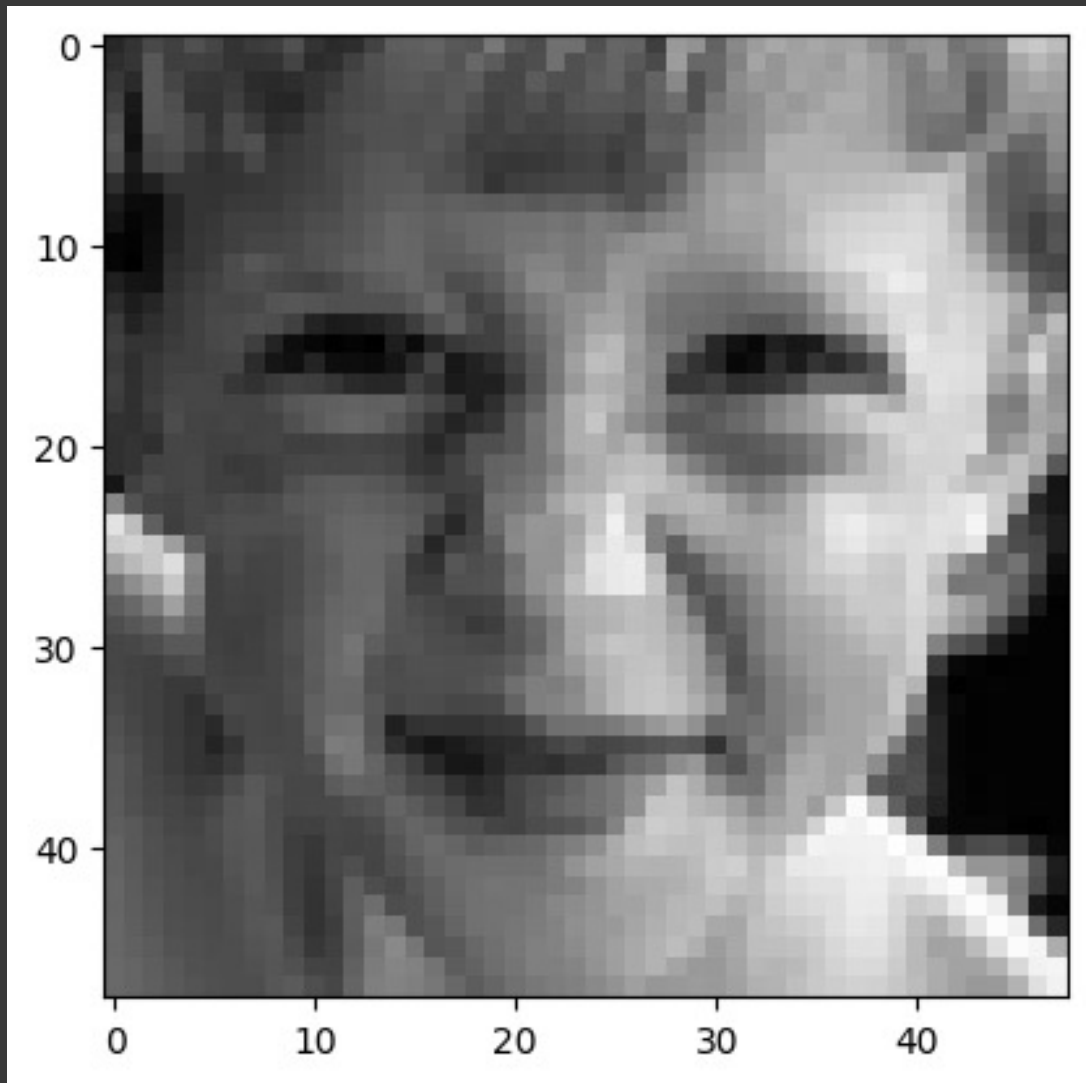
Adrian Sanz

Enric Tobeña

1 Data understanding



1 Data understanding



Las imágenes que componen el dataset son de 48x48 pixels en escala de grises de caras de personas

1 Data understanding

Train

```
[ ] train_generator = train_datagen.flow_from_directory(directory = train_dir,
                                                         target_size = (48,48),
                                                         batch_size = 64,
                                                         color_mode = "grayscale",
                                                         class_mode = "categorical",
                                                         subset = "training",
                                                         )
```

Found 20099 images belonging to 7 classes.

Test

```
[ ] validation_generator = test_datagen.flow_from_directory( directory = validation_dir,
                                                             target_size = (48,48),
                                                             batch_size = 64,
                                                             color_mode = "grayscale",
                                                             class_mode = "categorical",
                                                             subset = "validation",
                                                             shuffle=False
                                                             )
```

Found 6456 images belonging to 7 classes.

2 Data preprocessing - Train

```
# Creación del generador de aumento de datos
train_datagen = ImageDataGenerator(
    rescale=1./255,          # Escalado de los valores de los píxeles para que estén entre 0 y 1
    validation_split=0.3,    # Porcentaje de imágenes reservadas para validación
    rotation_range=10,       # Rango de rotación de la imagen en grados (-10 a 10)
    width_shift_range=0.1,   # Rango de desplazamiento horizontal aleatorio (-10% a 10% del ancho de la imagen)
    height_shift_range=0.2,  # Rango de desplazamiento vertical aleatorio (-20% a 20% de la altura de la imagen)
    zoom_range=0.2,          # Rango de zoom aleatorio
    shear_range=0.2,         # Rango de cizallamiento aleatorio (ángulo de cizallamiento: -20° a 20°)
    fill_mode='nearest'      # Estrategia de relleno de nuevos píxeles creados durante las transformaciones
)
```

2 Data preprocessing - Train

angry



fearful



disgusted



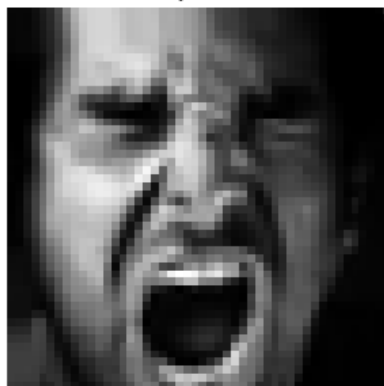
happy



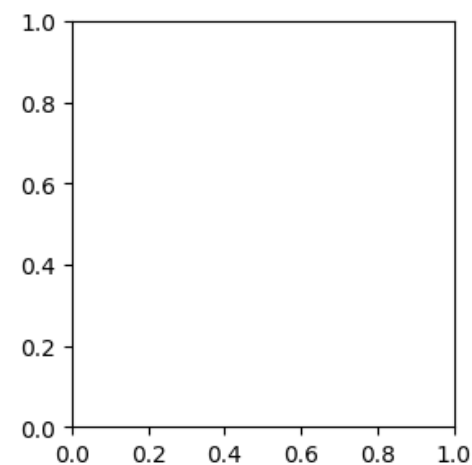
neutral



surprised



sad



2 Data preprocessing - Test

```
) test_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.9,  
    )
```

2 Data preprocessing - Test

```
) test_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.9,  
    )
```


3 Model 1

```
num_classes = len(ordered_class_names) # Número de clases de salida
input_shape = (48, 48, 1) # Formato de entrada

model1 = tf.keras.Sequential([

    # Capa convolucional 1
    tf.keras.layers.Conv2D(8, (3, 3), activation='relu', input_shape=input_shape),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),

    # Capa convolucional 2
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),

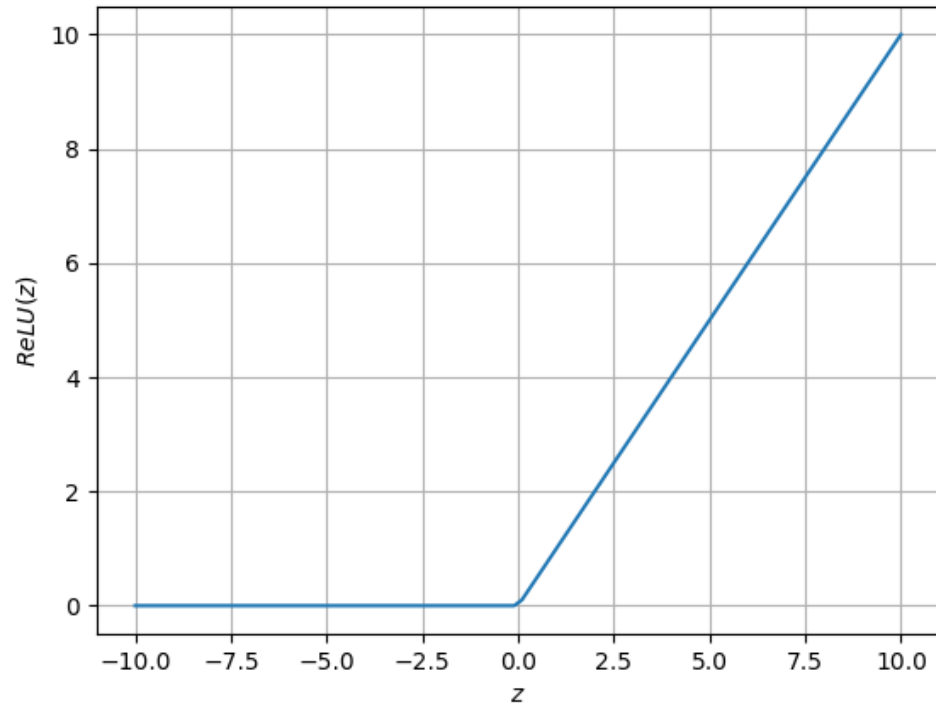
    # Capa convolucional 3
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),

    # Capa de aplanamiento (Flatten)
    tf.keras.layers.Flatten(),

    # Capa densa 1
    tf.keras.layers.Dense(64, activation='relu'),

    # Capa densa de salida
    tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

3 Model 2- Activation relu



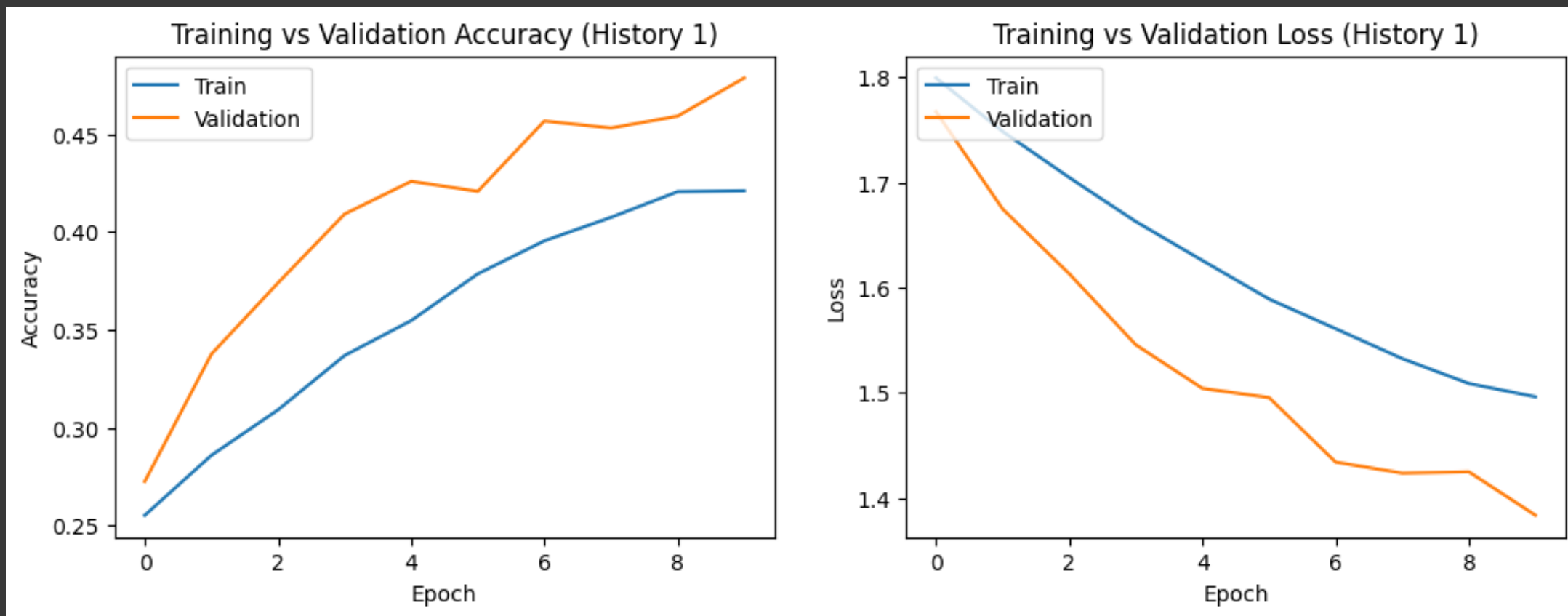
Función de activación Relu

3 Model 2 – Padding same

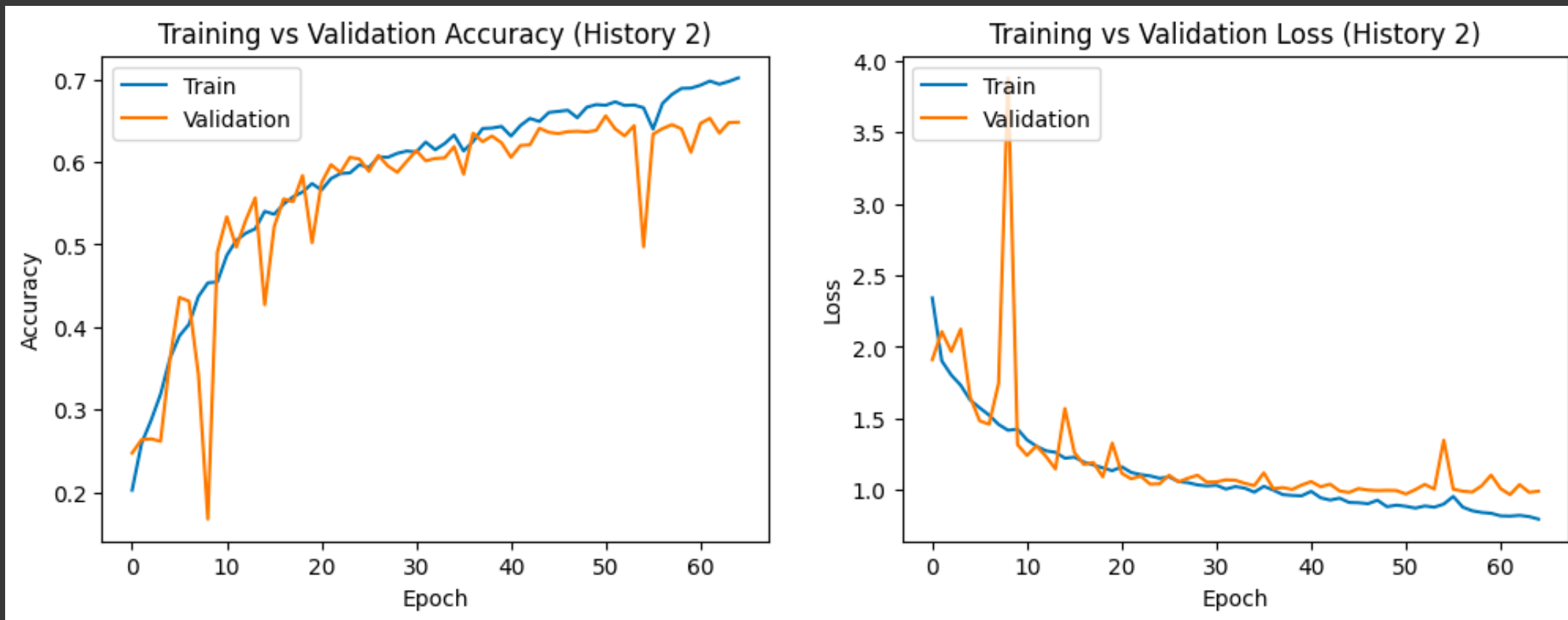


Ejemplo donde se tiene que obtener información también de las esquinas de la imagen

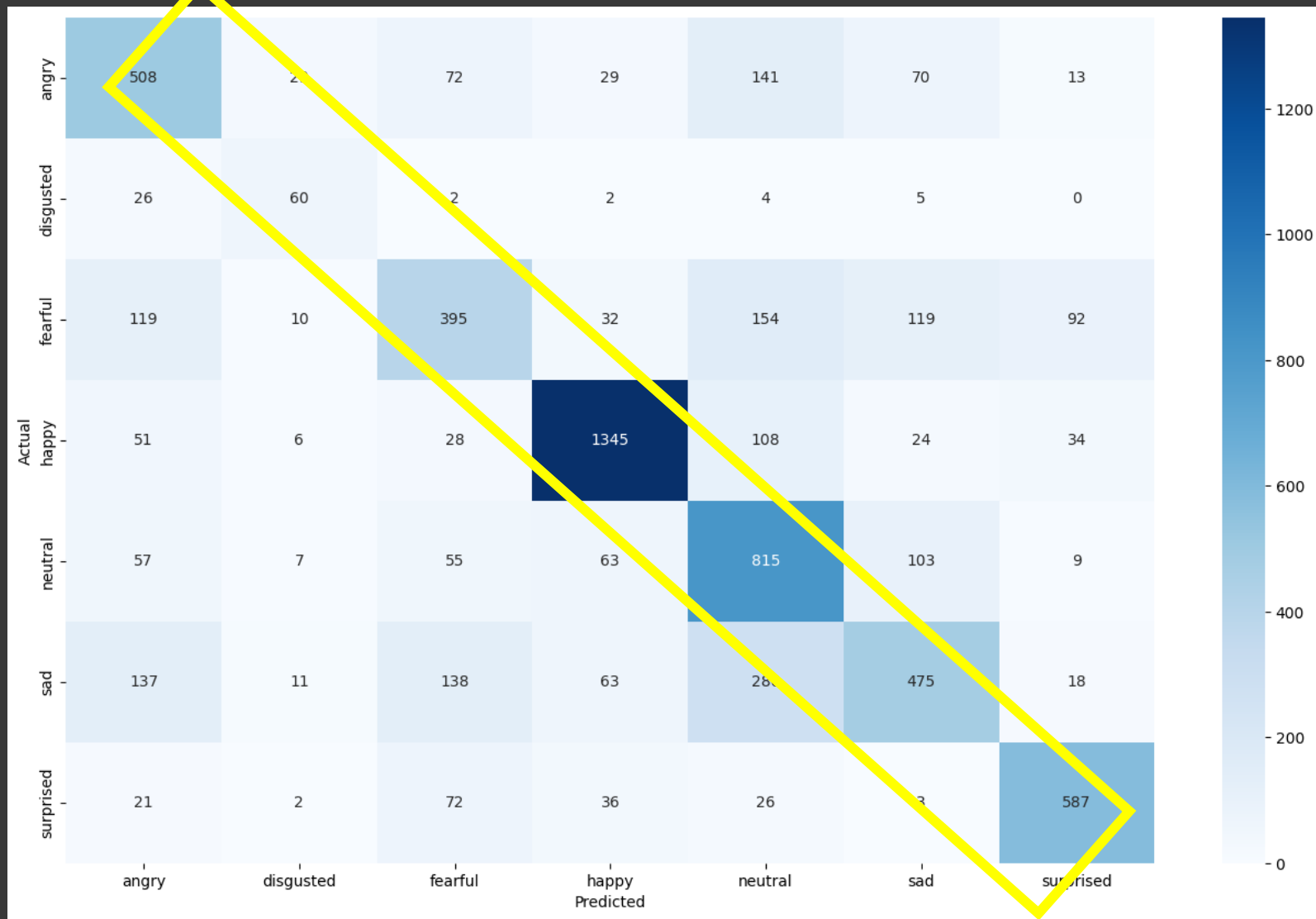
3 Results – Model 1



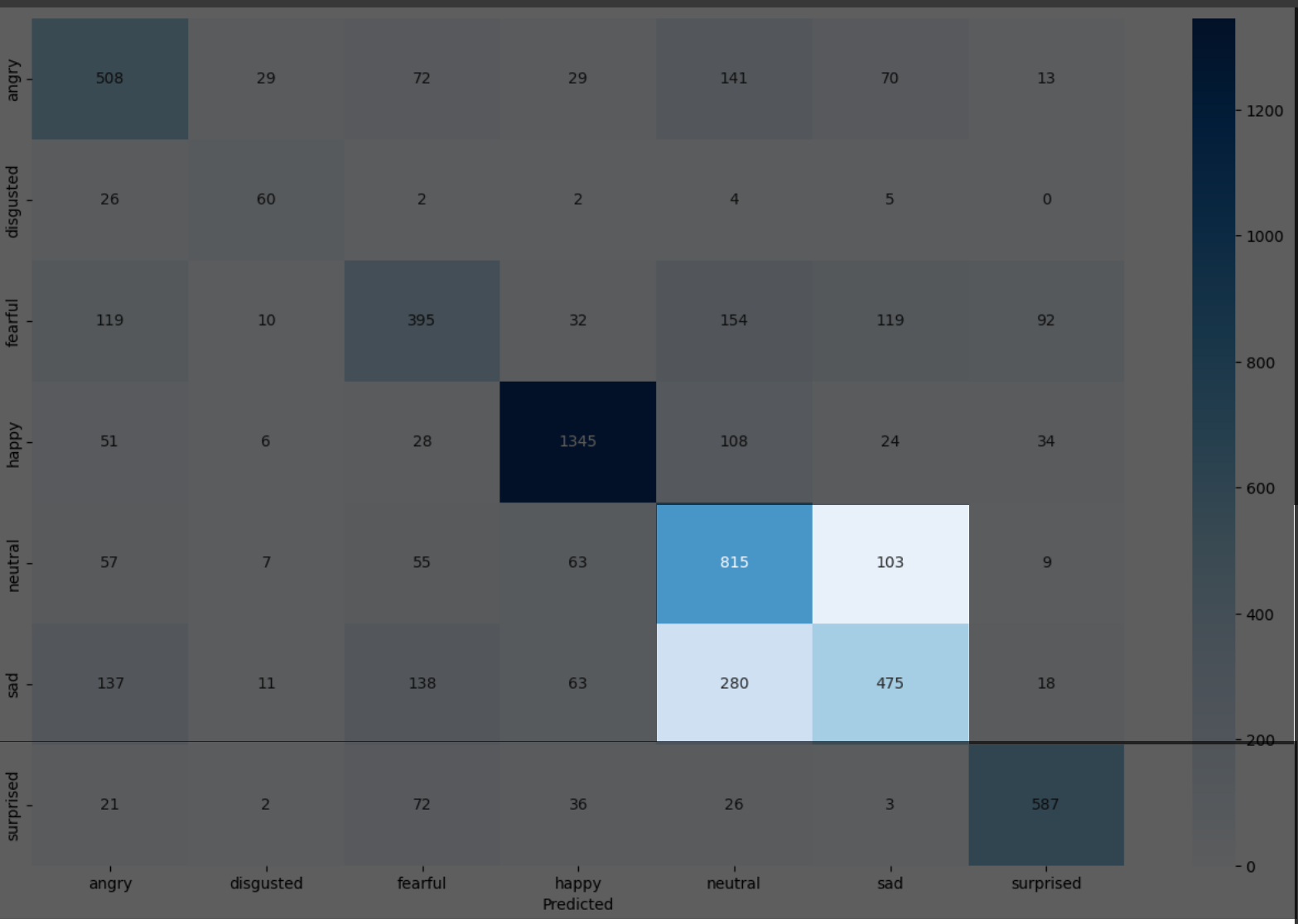
3 Results – Model 2



3 Results – Model 2



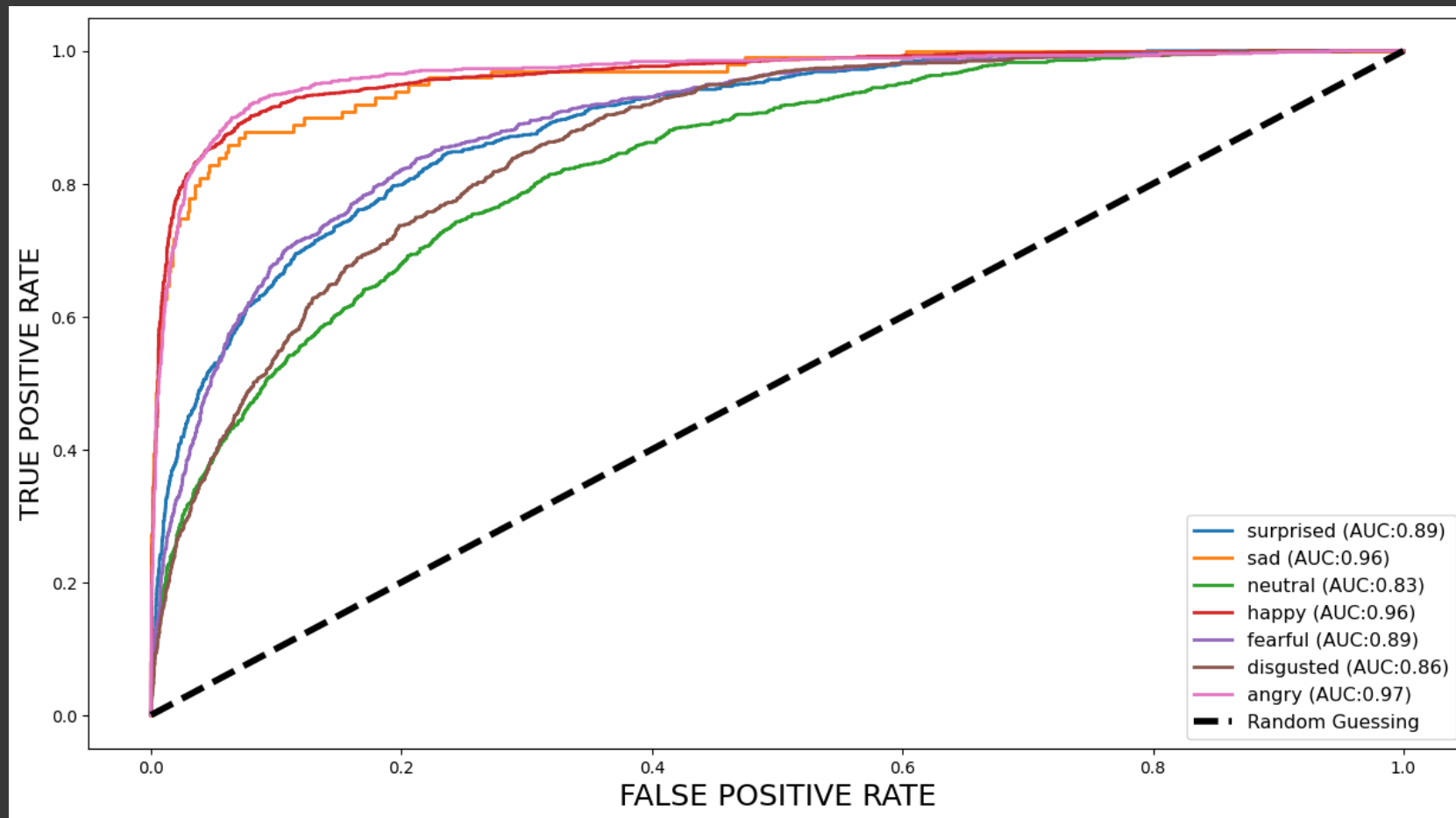
3 Results – Model 2



3 Results – Model 2

Una **curva ROC** (**curva de característica operativa del receptor**) es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación

ROC AUC score: 0.922695056841418



3 Results – Model 2

Una **curva ROC** (curva de característica operativa del receptor) es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación

ROC AUC score: 0.922695056841418

