Expositor

Joel Alberto Vilca Tarazona

+51 945440723

joelvilcatarazona@gmail.com

www.linkedin.com/in/vilcajoal/

Data Science

Tema

Sistema inteligente basado en Deep Learning para el proceso de clasificación de residuos

RESUMEN

Data Science

PROBLEMA

Inexistente automatización en el proceso de clasificación de residuos

Hogares no capacitados en buenas prácticas de clasificación de basura

Población genera cada vez más residuos.

Deficiente proceso en clasificación de residuos (papel, plástico, metal, cartón, vidrio y orgánico) en plantas de reciclaje.

Bajo porcentaje de basura reciclada

Menor impacto positivo al medio ambiente.

Mayor exposición del personal clasificador a enfermedades

Aumento de carga laboral para personal clasificador

Mayor exposición a lesiones en el proceso manual de clasificación

Mayor tiempo en proceso de clasificación de residuos

OBJETIVO

Entrenar un algoritmo de clasificación basado en Deep Learning

Realizar sugerencias para proyectos futuros utilizando el modelo

Evaluar y comparar los algoritmos de Deep Learning

Implementar un algoritmo utilizando redes neuronales convolucionales para clasificación de residuos reciclables (papel, plástico, metal, cartón, vidrio y orgánico).

Aumentar el porcentaje de basura reciclada

Aumentar impacto positivo al medio ambiente.

Minimizar exposición del personal clasificador a enfermedades

Disminuir carga laboral para personal clasificador

Minimizar exposición de lesiones en el proceso manual de clasificación

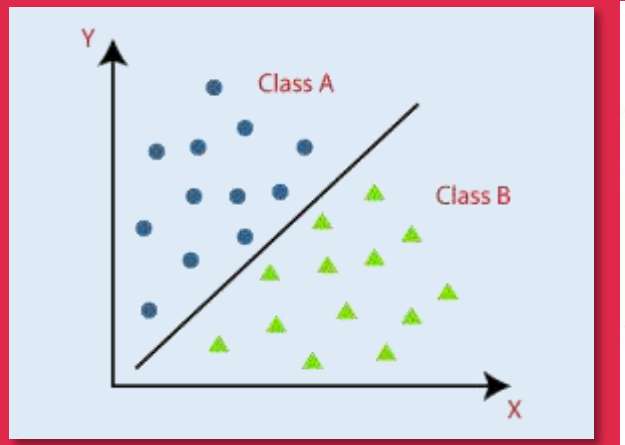
Minimizar tiempo en proceso de clasificación de residuos

MARCO TEÓRICO

Inteligencia Artificial

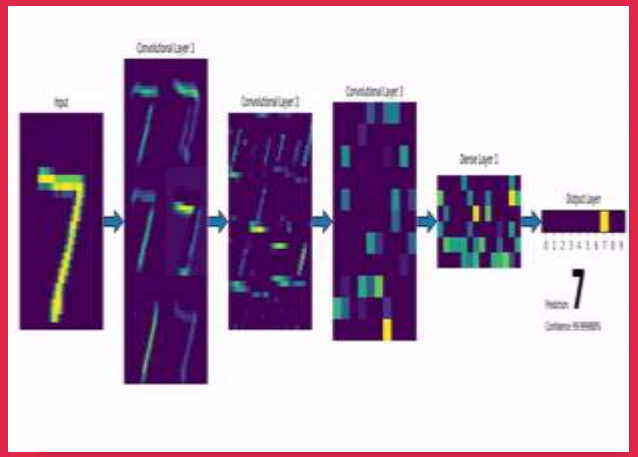
- Supervisado
- No supervisado

Algoritmos de clasificación



- Binario
- Multiclase

Redes Neuronales Convolucionales



- Convolución
- Detección de rasgos

Proceso de reciclaje

- Recolectar, acopiar
- clasificar, producir

ESTADO DEL ARTE

DEEP LEARNING PARA LA CLASIFICACIÓN Y SEGMENTACIÓN DE IMÁGENES DE ALIMENTOS

Iago Vidal Luna 2018

Importancia del Deep Learning y especificacmente las RNC a la hora de trabajar con imágenes el cual utilicé para implementación del algoritmo utilizando Python y Keras

Tesis

PROTOTIPO DE SISTEMA AUTOMATIZADO CON VISIÓN ARTIFICIAL PARA LA SELECCIÓN DE EMPAQUES DE PLÁSTICO, VIDRIO Y LATA EN EL PROCESO DE RECICLAJE

DIANA LUZ PACHÓN ESPINEL 2018

Amplio mi visión sobre las aplicaciones de deep learning en los procesos cotidianos y en la optimización de tareas repetitivas que me sirvió como guía para el algoritmo de clasificación que implementé.

Tesis

Classification for plastic bottles recycling based on image recognition

Zhaokun Wanga, Binbin Peng

Trabajo de clasificación de botellas de plástico en una cinta transportadora en al cual se estudia la influencia del tamaño de la muestra de entrenamiento en el modelo de clasificación y los resultados obtenido, me ayudó a entender que necesitaba buscar más imágenes para entrenamiento

Artículo

An Innovative Automated Robotic System based on Deep Learning approach for Recycling Objects

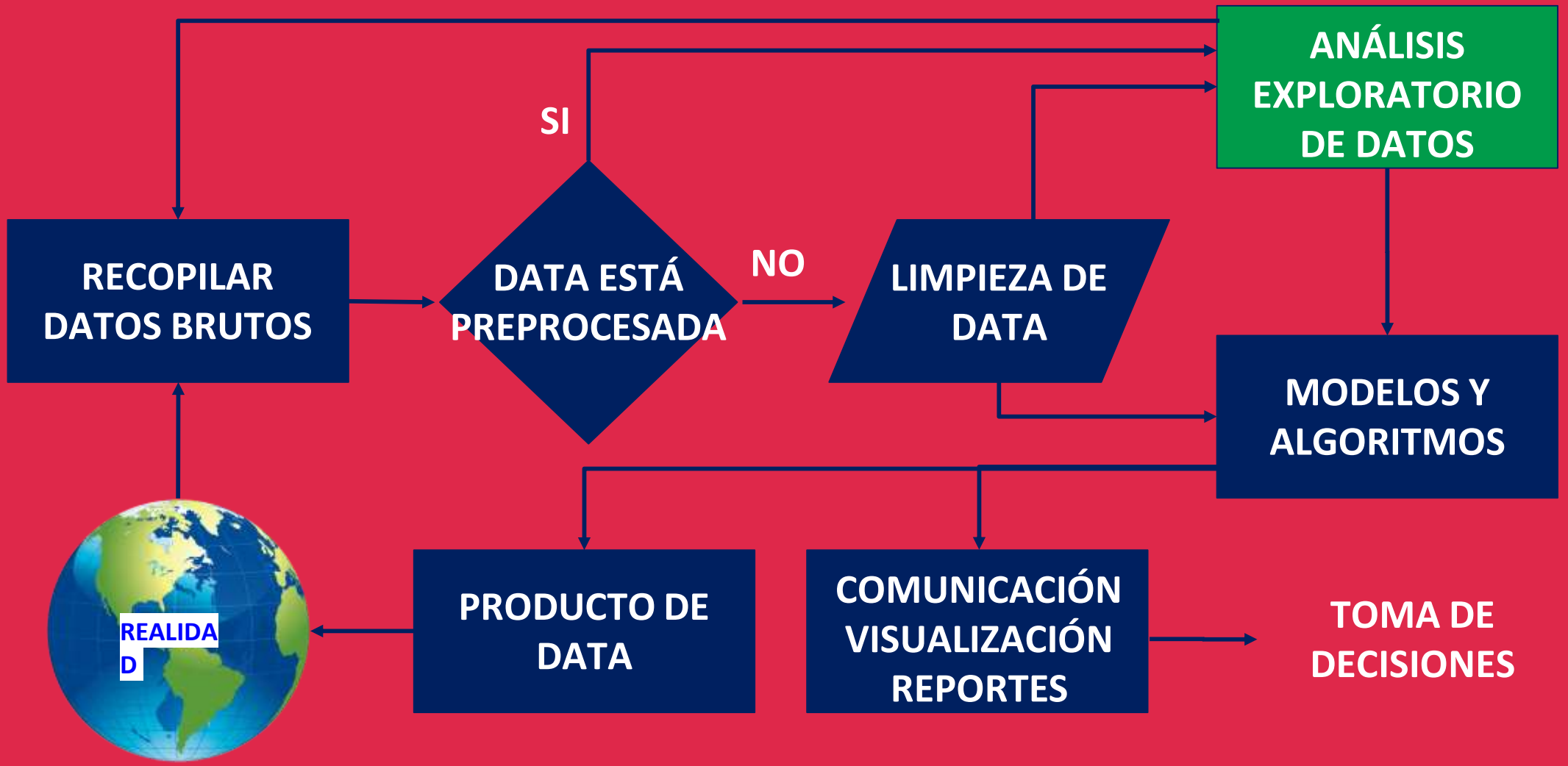
Jaeseok Kim, Olivia Nocentini , Marco Scafuro

Implementación de algoritmo de Deep Learning LeNet junto a un robot para clasificar residuos en plástico y cartón alcanzando precisión del 86.67%, me ayudó a investigar sobre RNC pre entrenadas como VGG16,ResNet50, InceptionResNetV2,Exception.

Artículo

METODOLOGÍA

DATA SCIENCE PROCESS



www.datascience.pe

DOING DATA SCIENCE(CATHY O'NEIL & RACHEL SCHUTT)

kaggle

trashnet

Código (solo para la red neuronal convolucional) y conjunto de datos para el proyecto final mío y de Mindy Yang para la clase CS 229: Machine Learning de Stanford . Nuestro artículo se puede encontrar aquí . Los resultados de la red neuronal convolucional en el póster están fechados ya que continuamos trabajando después del final del trimestre y pudimos lograr alrededor del 75% de precisión de la prueba (con 70/13/17 train / val / test split) después de cambiar la inicialización del peso a el método Kaiming.

Conjunto de datos

Este repositorio contiene el conjunto de datos que recopilamos. El conjunto de datos abarca seis clases: vidrio, papel, cartón, plástico, metal y basura. Actualmente, el conjunto de datos consta de 2527 imágenes:

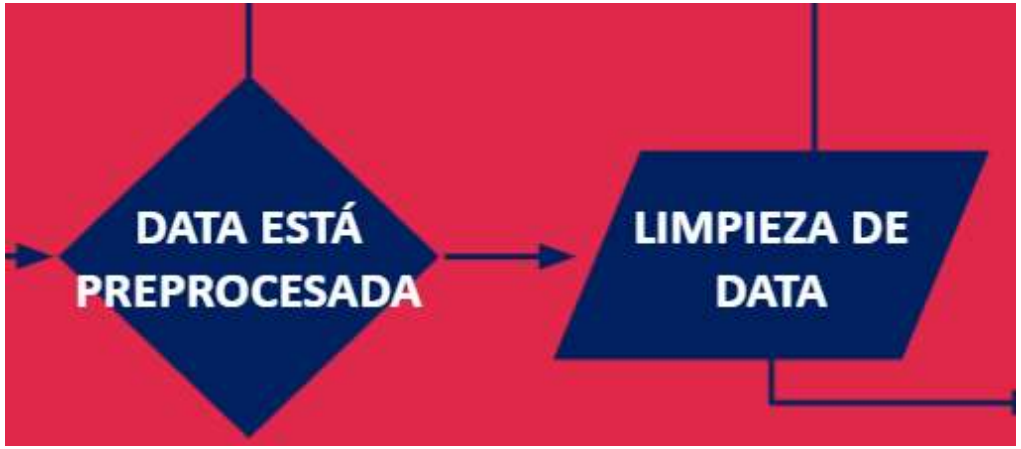
- 501 vidrio
- 594 papel
- 403 cartón
- 482 plástico
- 410 de metal
- 137 basura

Set de datos de 6,000 imágenes con 6 categorías

- 1,000 vidrio
- 1,000 papel
- 1,000 cartón
- 1,000 plástico
- 1,000 metal
- 1,000 orgánico

Data Science

```
for i in range(12):
    plt.subplot(2,6,i+1)
    plt.imshow(imgs[i])
    plt.title(rutas[i].split("\\")[-1])
    plt.axis("off")

plt.subplots_adjust(left=0.125,
                    bottom=0.1,
                    right=0.9,
                    top=0.4,
                    wspace=0.2,
                    hspace=0.35)

plt.show()
```




```
# imagenes reesaladas a 224x224
img_width = 224
img_height = 224
batch_size = 100
```

```
# aumento y normalizacion de datos
datagen_train = ImageDataGenerator(rescale=1.0/255.0,      # Normalizar imagenes en range [0-1]
                                   horizontal_flip=True,   # Volteo horizontal
                                   rotation_range=15,      # Rotación aleatoria entre 0 y 15 grados
                                   width_shift_range=0.15,  # Mover la imagen horizontalmente 15%
                                   height_shift_range=0.15, # Mover imagen verticalmente 15%
                                   zoom_range=0.2)          # Acercar / Alejar aleatoriamente 20% => [80% - 120%]
```

```
# leer imagenes de carpeta conjunto train
training_set_imgs = datagen_train.flow_from_directory(project_folder + '/train',
                                                      target_size = (img_width, img_height),
                                                      class_mode = 'categorical',
                                                      batch_size = batch_size)
```

Science

Fase de Modelado


```
for layer in incep_conv.layers[:]:
    layer.trainable = False
```

```
for layer in incep_conv.layers:
    print(layer, layer.trainable)
```

```
# Model: "inception_resnet_v2"
incep_conv = InceptionResNetV2(weights='imagenet', include_top=False, input_shape=(img_width, img_height, 3))
incep_conv.summary();
```

```
model = Sequential()

model.add(incep_conv)

model.add(Flatten())

model.add(Dense(4096, activation='relu'))
model.add(Dropout(rate=0.4))
model.add(Dense(4096, activation='relu'))
model.add(Dropout(rate=0.4))
model.add(Dense(6, activation='softmax'))
```

```
opt = Adam(lr=0.0001, beta_1=0.9, beta_2=0.999, amsgrad=False)
#opt = SGD(lr=0.0001, momentum=0.9)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
```

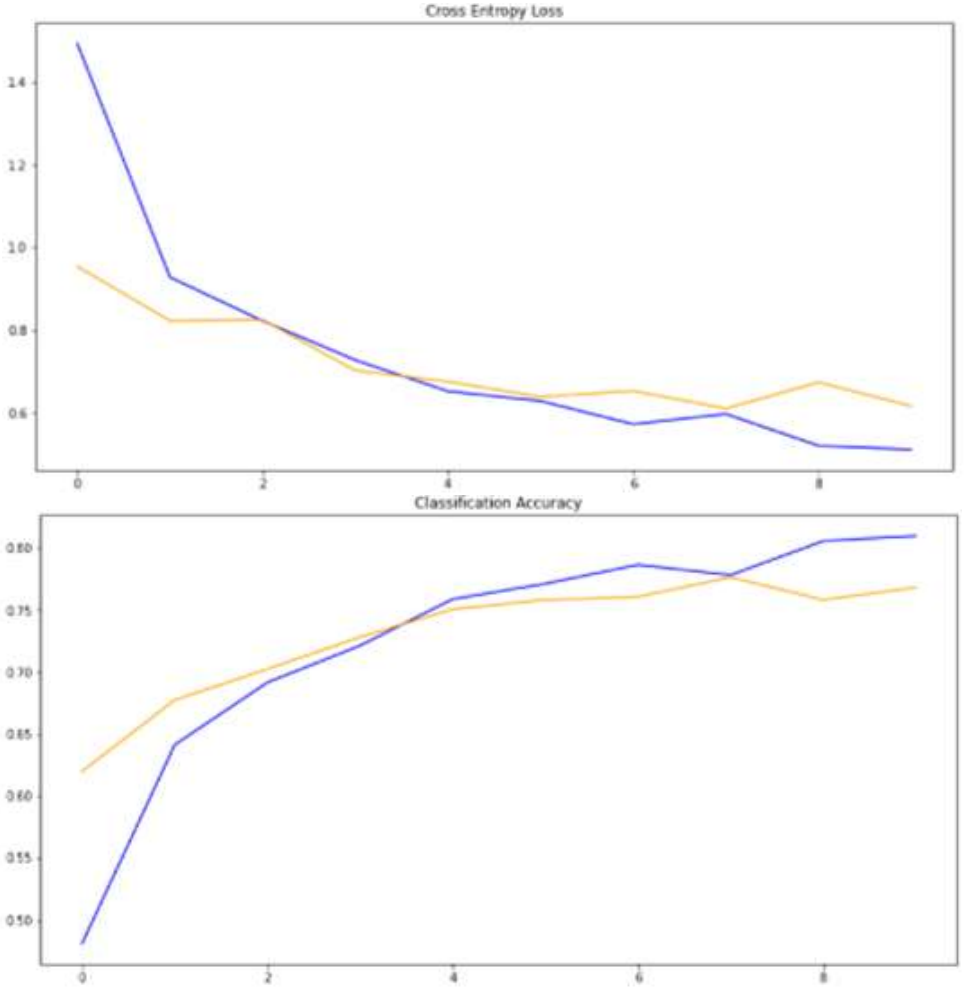
```
epochs=10
history = model.fit_generator(training_set_imgs,
                              epochs=epochs,
                              steps_per_epoch=np.ceil(num_imgs_training/batch_size),
                              validation_data=valid_set_imgs,
                              validation_steps=np.ceil(num_imgs_valid/batch_size))
```

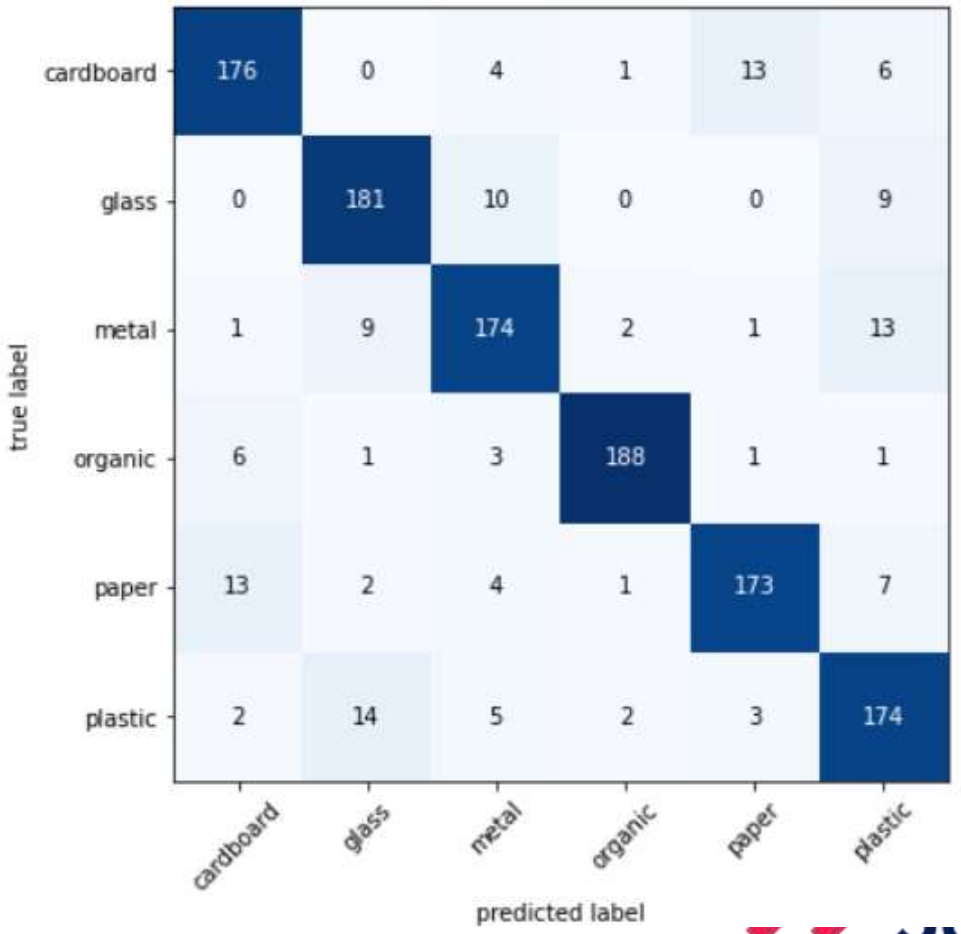
Data Science

Fase de Evaluación

MODELOS Y ALGORITMOS

```
Validación en Test:
Loss: 0.3792
Accuracy: 88.08%
```

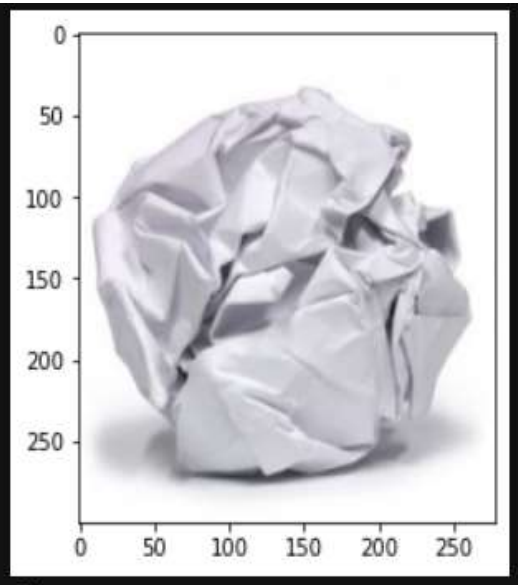




Guardar modelo y predecir


```
# Keras formato
carperta_proyecto = "F:\Repositorios\DSRP_Semillero_ClasificacionResiduos"
model.save(carperta_proyecto +"\Modelos\model_vgg16.h5", save_format='h5')
print("Modelo guardado en disco ...")
```

```
loaded_model = load_model(carperta_proyecto + "\Modelos\model_vgg16.h5")
loaded_model.summary()
```



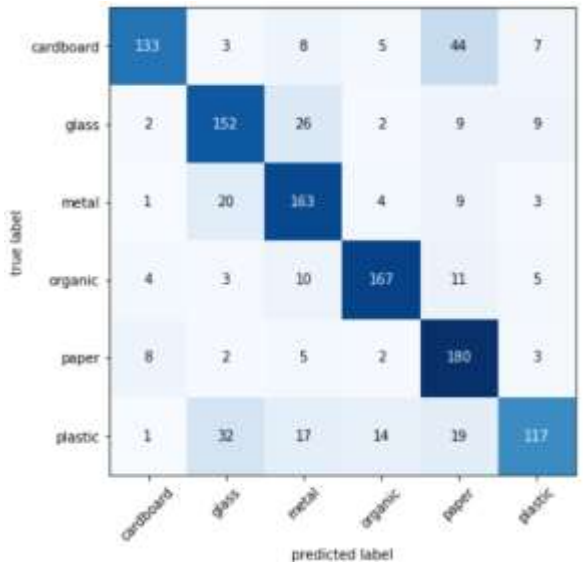

```
[0.01010183 0.00128938 0.0357519  0.00088975 0.7766795  0.17528763]
Index: 4
Pedicción: paper
Prob: 0.7766795
```

```
folder_ejemplos = carperta_proyecto+"/imgs"
img_ruta = folder_ejemplos + '/ejemplo_papel.jpg'
test_image = image.load_img(img_ruta)
plt.imshow(test_image, aspect="auto")
plt.show()

test_image = image.load_img(img_ruta,target_size = (img_width, img_height))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
test_image = test_image.astype('float32')
test_image /= 255.0

predictions = loaded_model.predict(test_image)[0]
print(predictions)
index = np.argmax(predictions)

CLASSES =  [ 'cardboard','glass', 'metal', 'organic', 'paper', 'plastic']
ClassPred = CLASSES[index]
ClassProb = predictions[index]

print("Index:", index)
print("Pedicción:", ClassPred)
print("Prob:", ClassProb)
```

Comparación de modelos

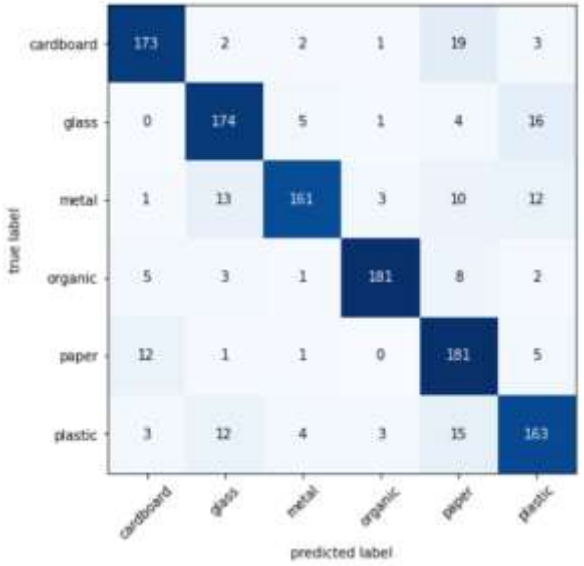
MODELOS Y ALGORITMOS

VGG16



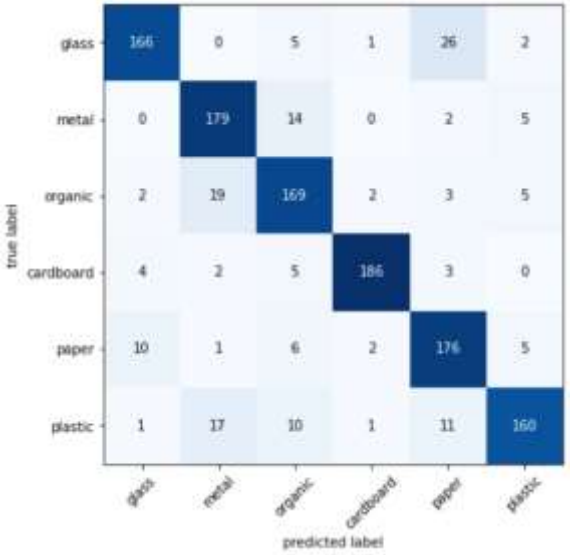
Accuracy: 76.67%

ResNet101V2



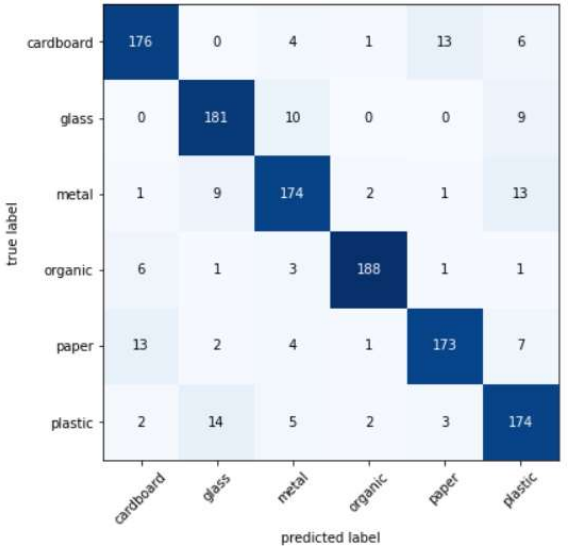
Accuracy: 87.50%

InceptionResNetV2



Accuracy: 86.67%

DenseNet201



Accuracy: 88.08%

Data Science

TRABAJOS A FUTURO

PRODUCTO DE DATA

COMUNICACIÓN VISUALIZACIÓN REPORTES

TOMA DE DECISIONES

www.datascience.pe

Data Science

¡Muchas gracias!