

Desafío 2 - Grupo 4



Trabajando con CNNs o RNNs

Carlos Escovino | Diego Lesertesseur | Emiliano Llera do Campo

Introducción

En este segundo desafío, deberán elegir un problema en concreto a resolver y proponer un dataset acorde al ejercicio.

Para esto, podrán utilizar algún dataset preexistente o construir uno propio recolectando datos a partir de herramientas de web scraping o consumo de APIs.

Objetivos

- Entrenar y optimizar una red neuronal convolucional que sea capaz de resolver un problema de computer vision, por ejemplo, clasificación de imágenes .
- Entrenar y optimizar una red neuronal recurrente que sea capaz de realizar alguna predicción en base a información secuencial, por ejemplo, pronóstico de series de tiempo o clasificación de textos.

Planteo general de desafío

Dataset

Seleccionar un dataset de imágenes categorizadas.

Modelos

Determinar una lista de modelos de redes convolucionales.

Hacer pruebas con dichos modelos.

Comparativa

Determinar cuál de los modelos probados tiene mejor precisión al momento de clasificar una imagen.

Implementación

Dataset

Stanford Dogs Dataset:

Este conjunto de datos se ha creado utilizando imágenes y anotaciones de ImageNet para la tarea de categorización de imagen detallada.

- Categorías: 120
 - Imágenes: 20,580
 - Anotaciones: etiquetas de clase, cuadros delimitadores
 - 757 MB
-

Modelos CNN

- Propio
 - Pre-Entrenado VGG19/RAdam
 - Pre-Entrenado InceptionV3/RAdam
 - Pre-Entrenado InceptionV3
 - Pre-Entrenado InceptionResNetV2
 - Pre-Entrenado NASNet/RAdam
 - Pre-Entrenado Xception
 - Pre-Entrenado VGG19/RAdam/Fine Tunning
 - Pre-Entrenado InceptionV3/RAdam/Fine Tunning
 - Pre-Entrenado NASNet/RAdam/Fine Tunning
-

Proceso de pruebas

- Se separa un porcentaje de las imágenes de cada categoría en train, test y validación.

Modelo CNN Propio - Arquitectura

Capa	Activación
Conv2D (32)	relu
MaxPooling2D	
Conv2D (64)	relu
MaxPooling2D	
Conv2D (128)	relu
MaxPooling2D	
Conv2D (128)	relu

Capa	Activación
Flatten	
Dense (512)	relu
Dense (120)	softmax

Kernel	3 x 3
pool_size	2 x 2

Modelo CNN Propio - Compilación

loss	categorical_crossentropy
optimizer	<p>RAdam</p> <ul style="list-style-type: none">• total_steps = 5000• warmup_proportion = 0.1• min_lr = 1e-7
metrics	categorical_accuracy

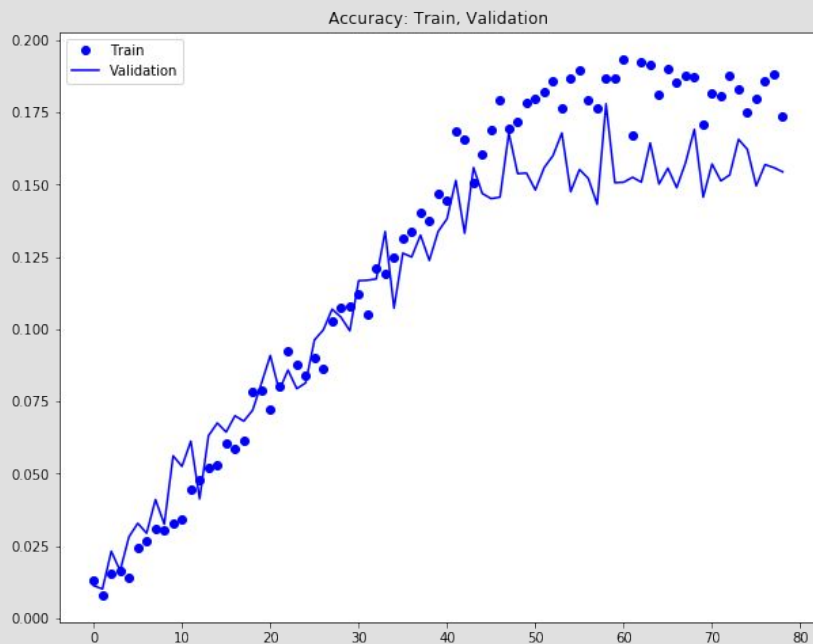
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

Modelo CNN Propio - Entrenamiento

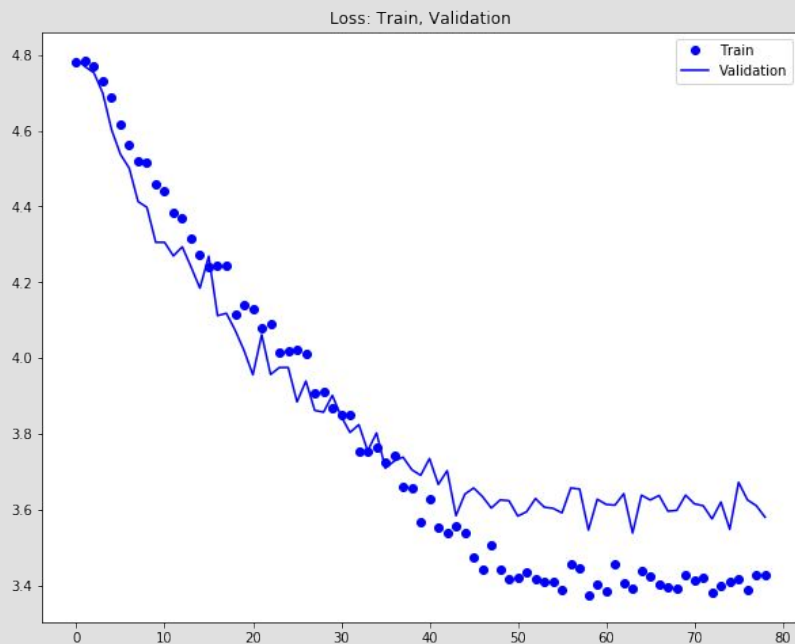
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 500• validation_data = valid_generator• validation_steps = 50

Modelo CNN Propio - Resultados

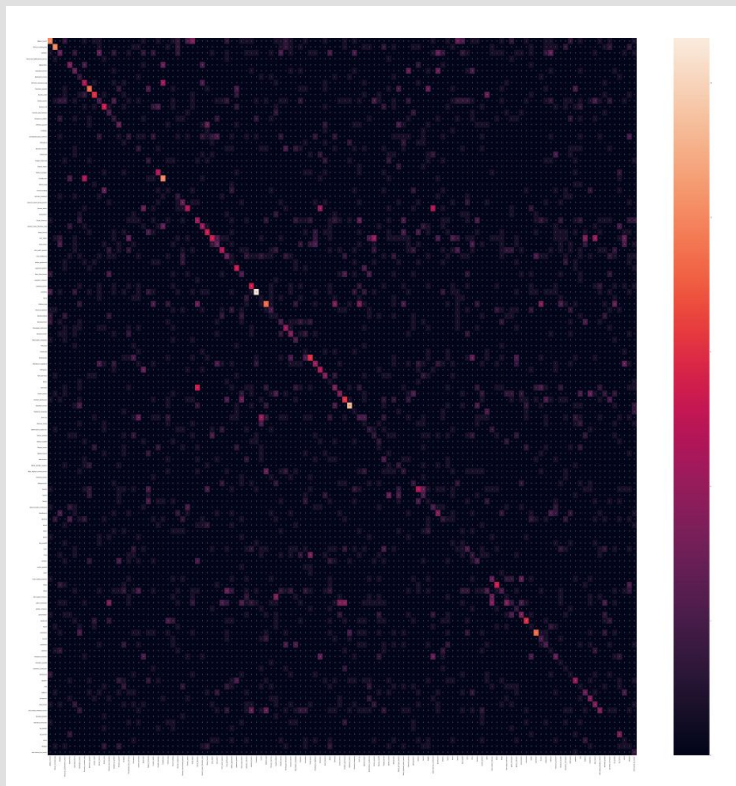
Precisión



Pérdida



Modelo CNN Propio - Matriz de confusión



Modelos Pre Entrenados

Modelo Pre Entrenado VGG19 / RAdam

VGG19 / RAdam - Arquitectura

Capa	Activación
conv_base (VGG19)	
Flatten	
Dense (256)	relu
Dense (120)	softmax

VGG19 / RAdam - Compilación

loss	categorical_crossentropy
optimizer	RAdam <ul style="list-style-type: none">• lr=1e-4
metrics	categorical_accuracy

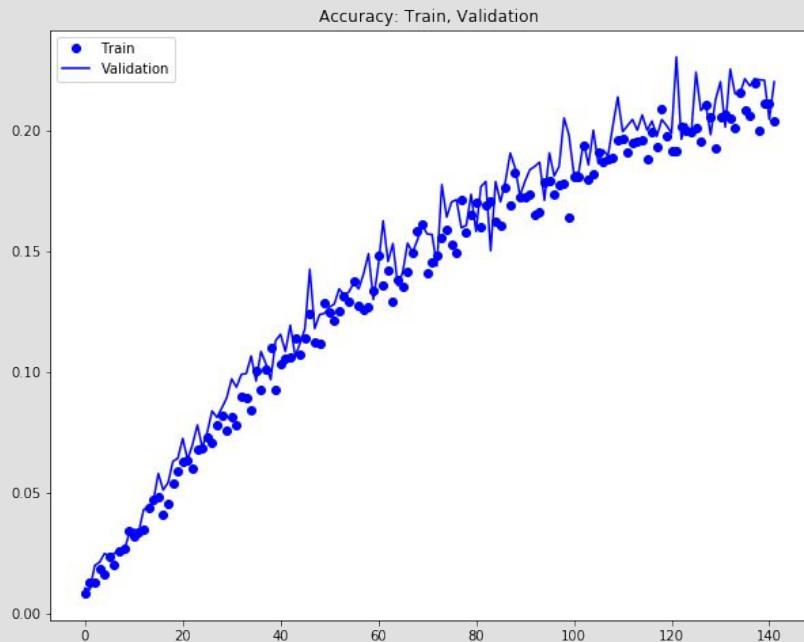
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

VGG19 / RAdam - Entrenamiento

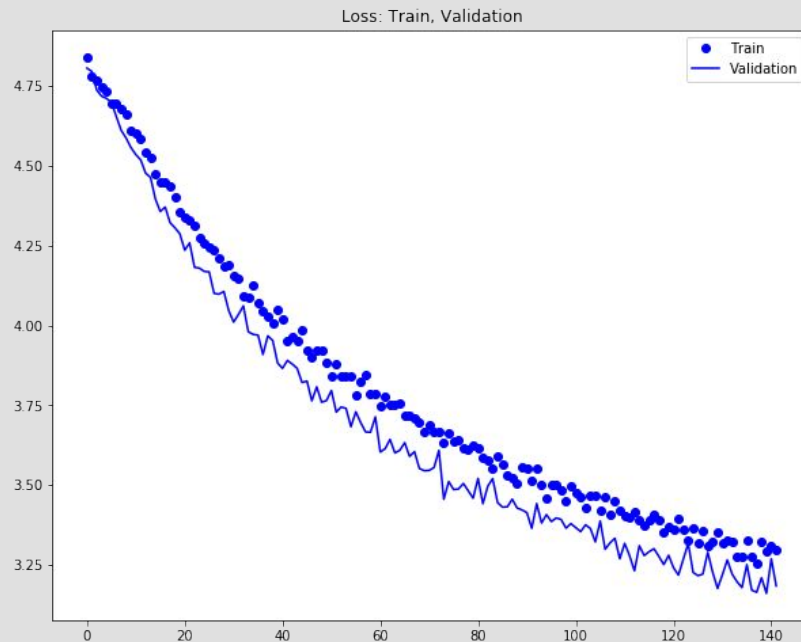
callbacks	<div>EarlyStopping</div> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 500• validation_data = valid_generator• validation_steps = 50

VGG19 / RAdam - Resultados

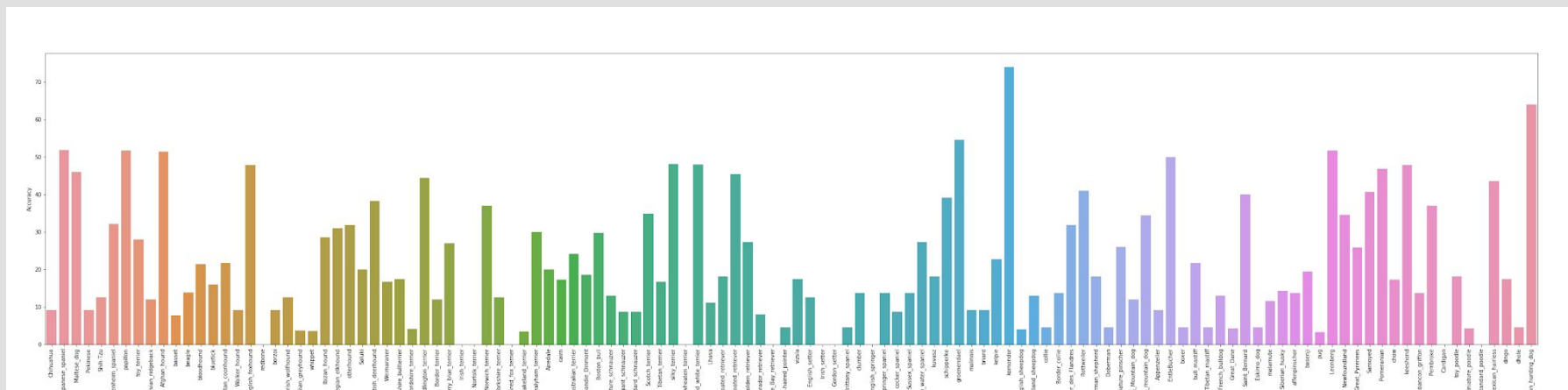
Precisión



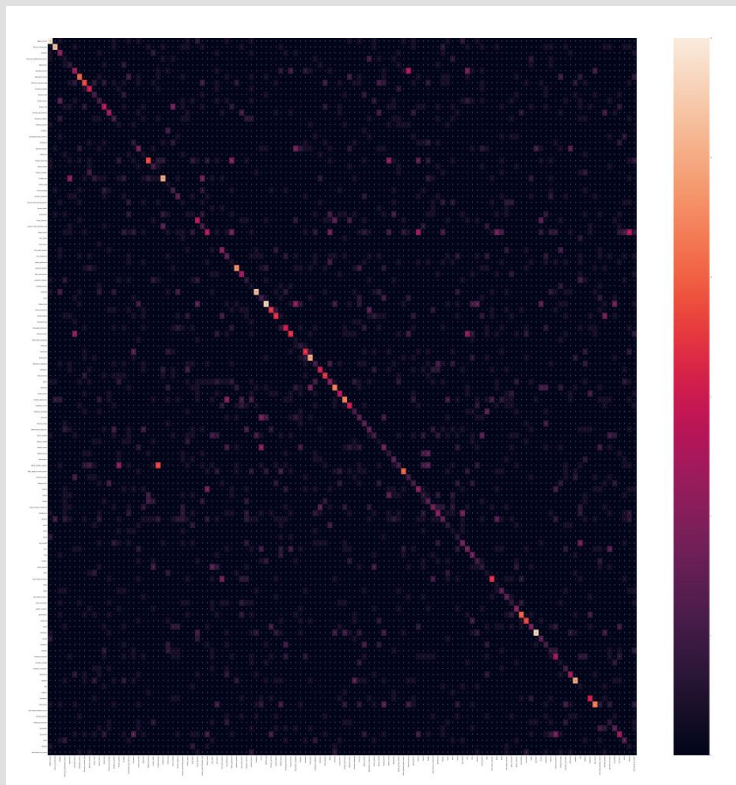
Pérdida



VGG19 / RAdam - Accuracy por categoría



VGG19 / RAdam - Matriz de confusión



Modelo Pre Entrenado InceptionV3 / RAdam

InceptionV3 / RAdam - Arquitectura

Capa	Activación
conv_base (inception_v3)	
Flatten	
Dense (256)	relu
Dense (120)	softmax

InceptionV3 / RAdam - Compilación

loss	categorical_crossentropy
optimizer	RAdam <ul style="list-style-type: none">• total_steps = 5000• warmup_proportion = 0.1• min_lr = 1e-7
metrics	categorical_accuracy

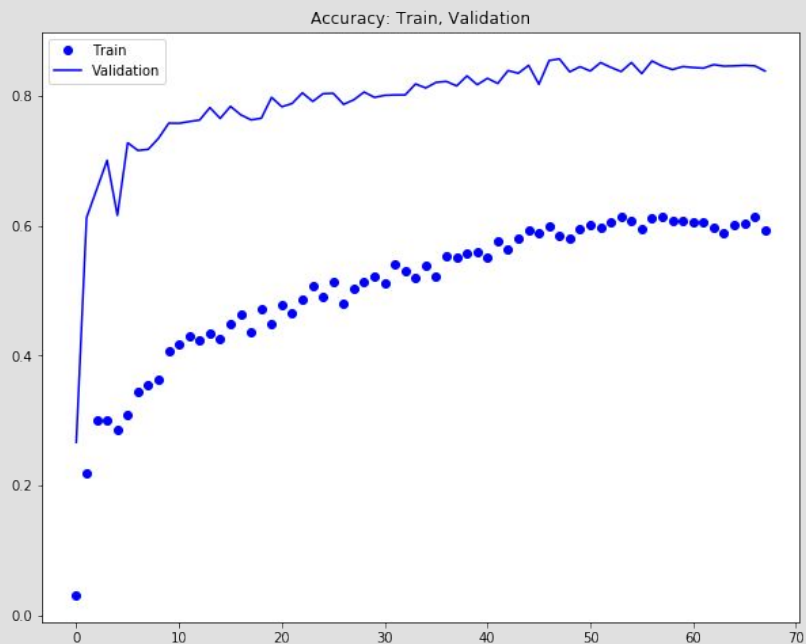
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

InceptionV3 / RAdam - Entrenamiento

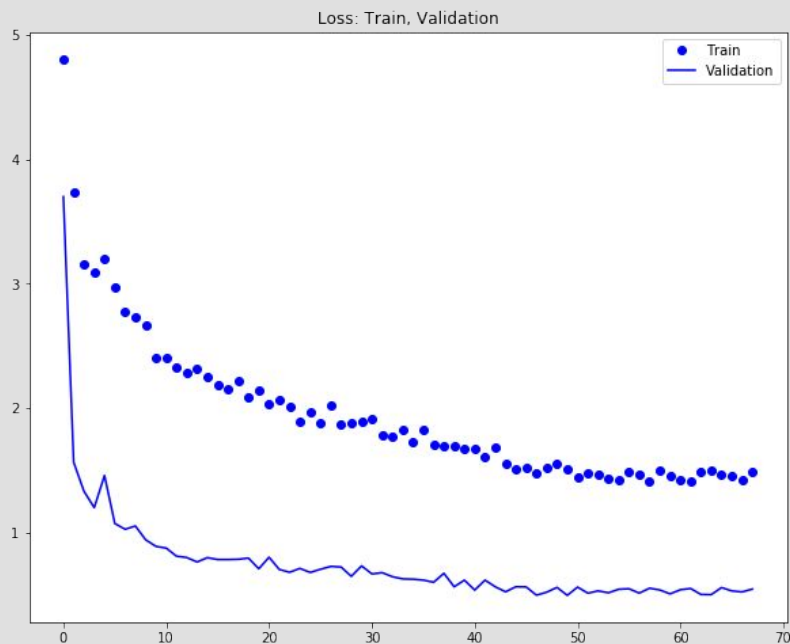
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 500• validation_data = valid_generator• validation_steps = 50

InceptionV3 / RAdam - Resultados

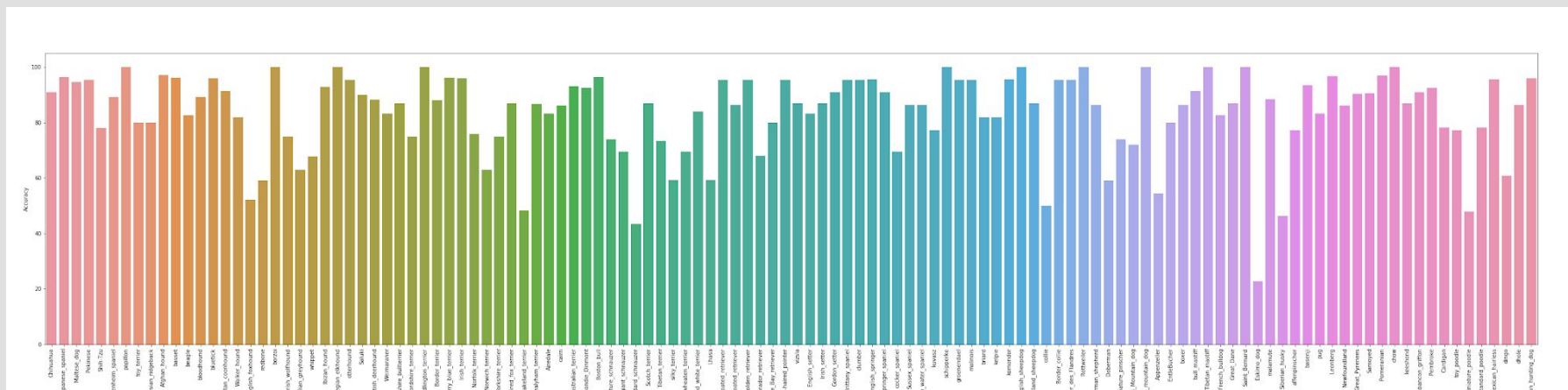
Precisión



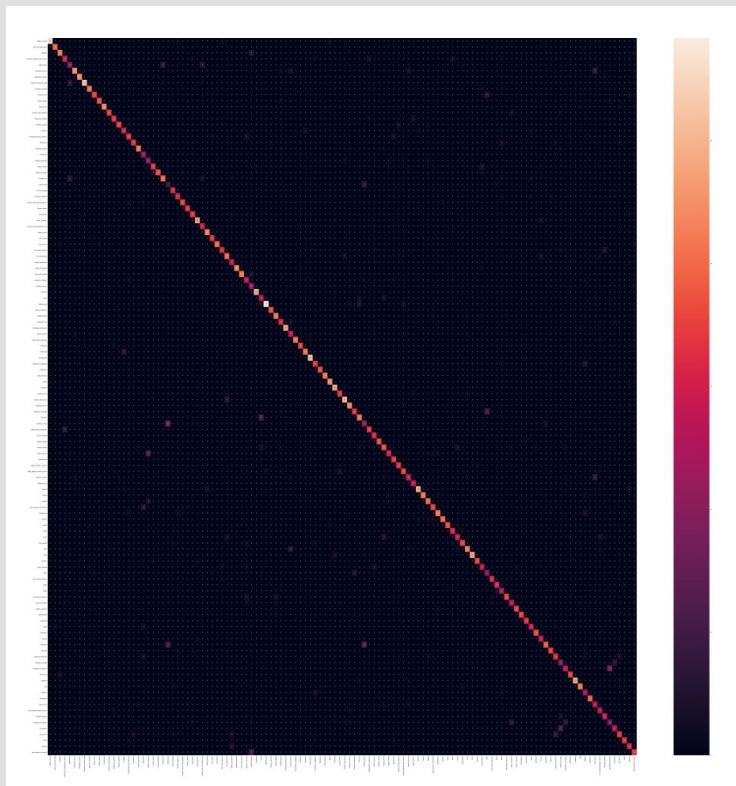
Pérdida



InceptionV3 / RAdam - Accuracy por categoría



InceptionV3 / RAdam - Matriz de confusión



Modelo Pre Entrenado InceptionV3

InceptionV3 - Arquitectura

Capa	Activación
conv_base (inception_v3)	
Flatten	
Dense (256)	relu
Dense (120)	softmax

InceptionV3 - Compilación

loss	categorical_crossentropy
optimizer	<div>RMSprop<ul style="list-style-type: none">• Lr = 1e-4</div>
metrics	categorical_accuracy

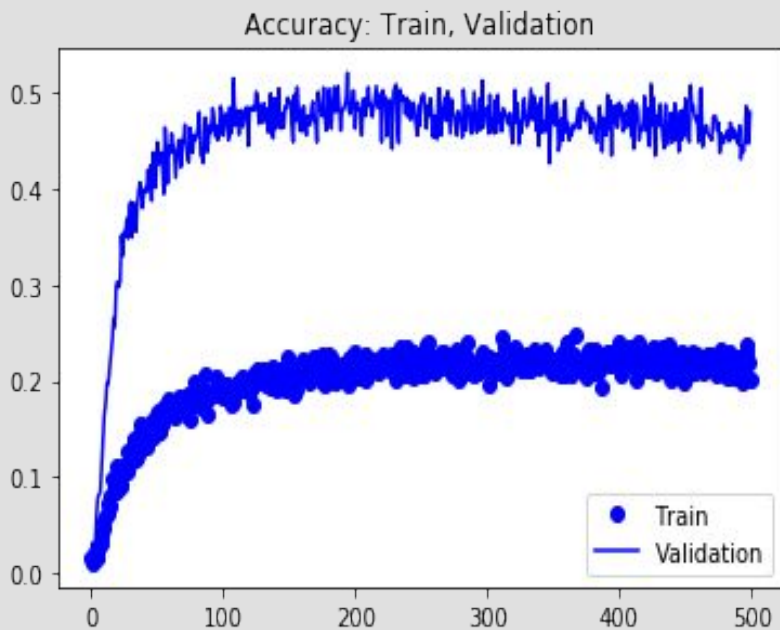
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

InceptionV3 - Entrenamiento

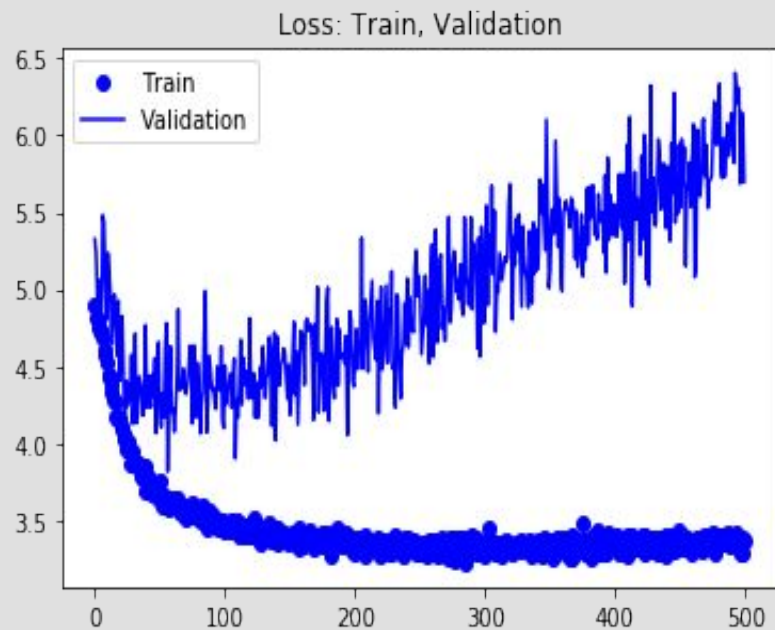
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 500• validation_data = valid_generator• validation_steps = 50

InceptionV3 - Resultados

Precisión



Pérdida



Modelo Pre Entrenado InceptionResNetV2

InceptionResNetV2 - Arquitectura

Capa	Activación
conv_base (inception_resnet_v2)	
Flatten	
Dense (256)	relu
Dense (120)	softmax

InceptionResNetV2 - Compilación

loss	categorical_crossentropy
optimizer	<p>RAdam</p> <ul style="list-style-type: none">• total_steps = 5000• warmup_proportion = 0.1• min_lr = 1e-7
metrics	categorical_accuracy

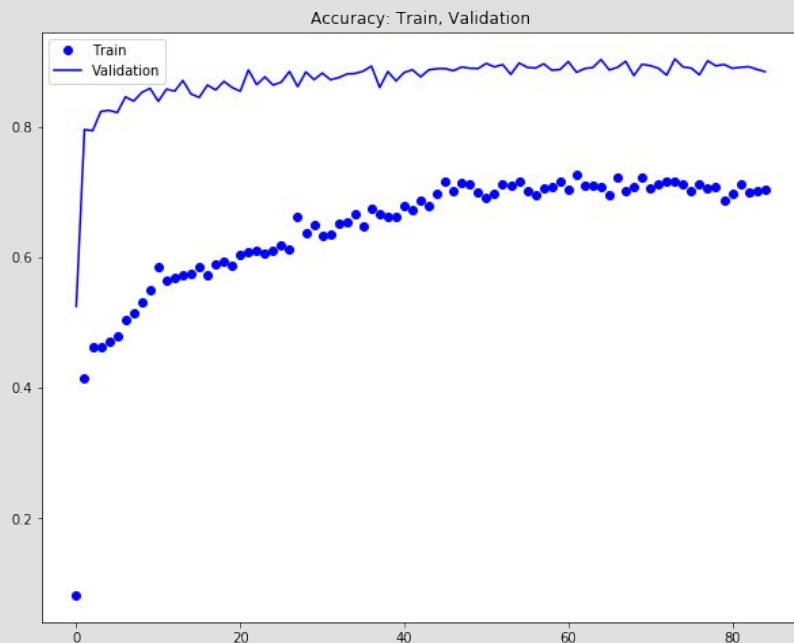
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

InceptionResNetV2 - Entrenamiento

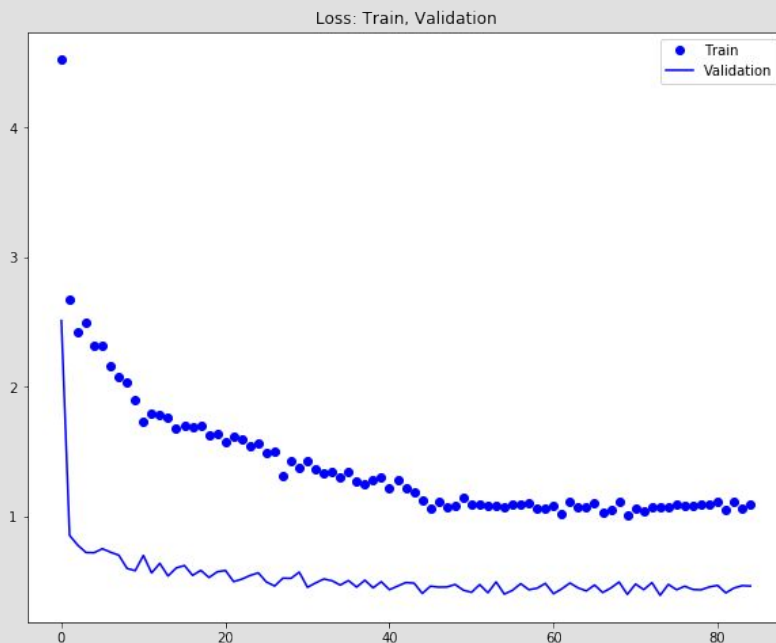
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 500• validation_data = valid_generator• validation_steps = 50

InceptionResNetV2 - Resultados

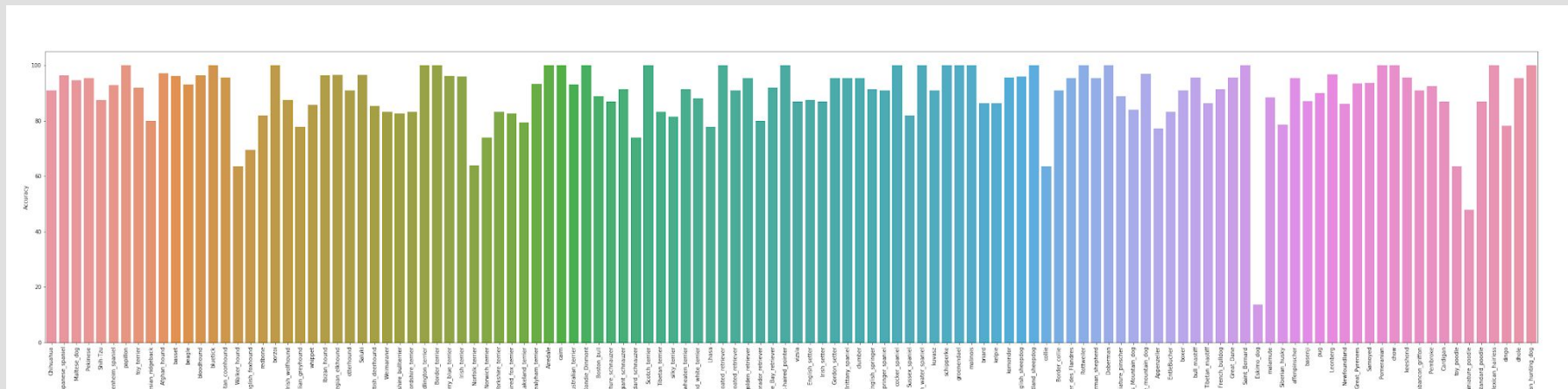
Precisión



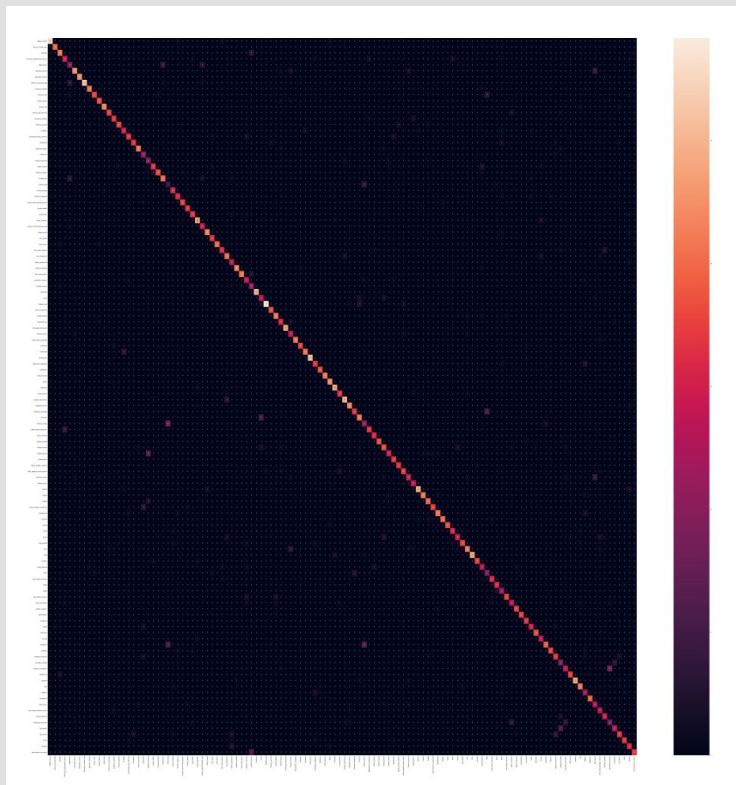
Pérdida



InceptionResNetV2 - Accuracy por categoría



InceptionResNetV2 - Matriz de confusión



Modelo Pre Entrenado NASNet

NASNet - Arquitectura

Capa	Activación
conv_base (NASNet)	
Flatten	
Dense (256)	relu
Dense (120)	softmax

NASNet - Compilación

loss	categorical_crossentropy
optimizer	<p>RAdam</p> <ul style="list-style-type: none">• total_steps = 5000• warmup_proportion = 0.1• min_lr = 1e-7
metrics	categorical_accuracy

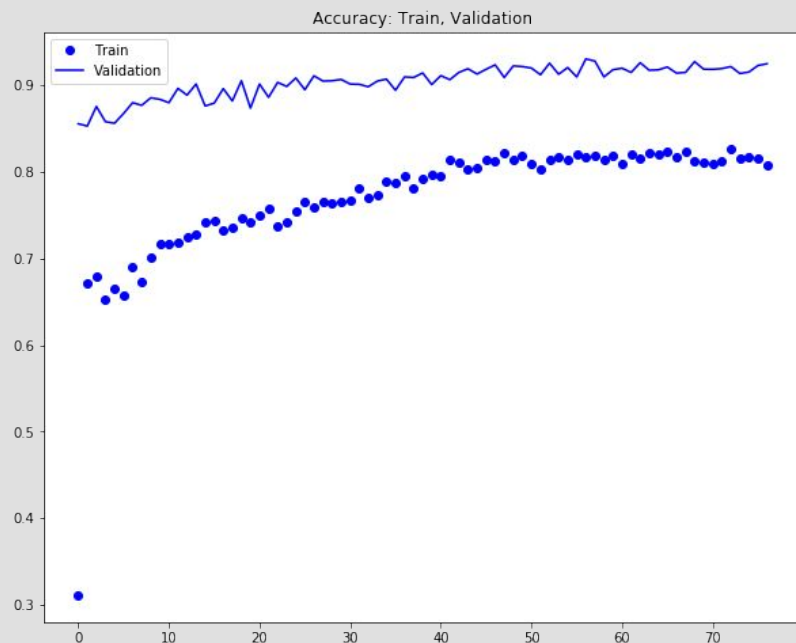
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

NASNet - Entrenamiento

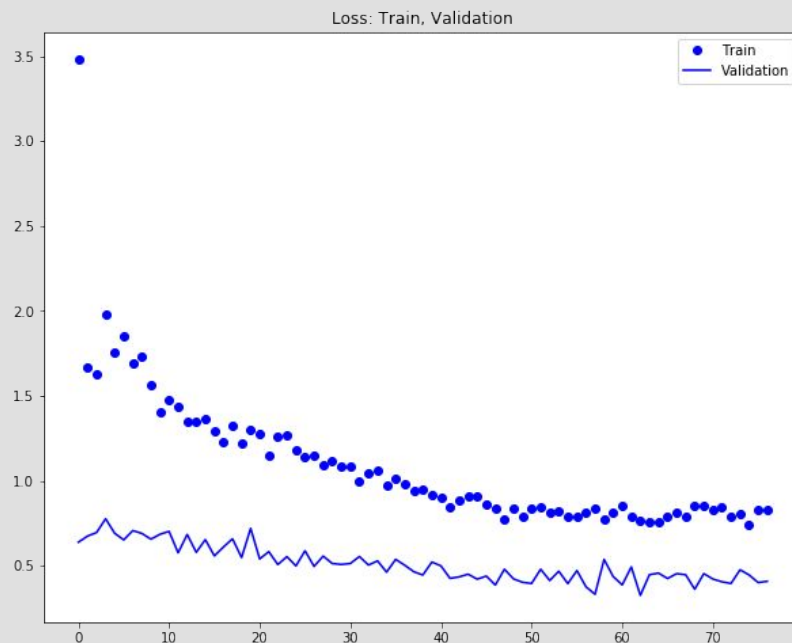
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 500• validation_data = valid_generator• validation_steps = 50

NASNet - Resultados

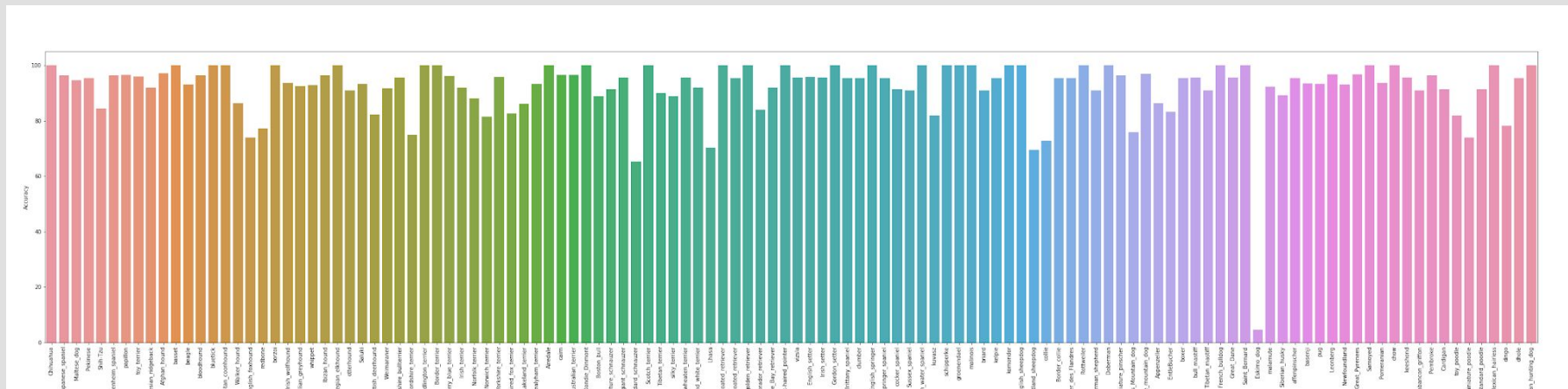
Precisión



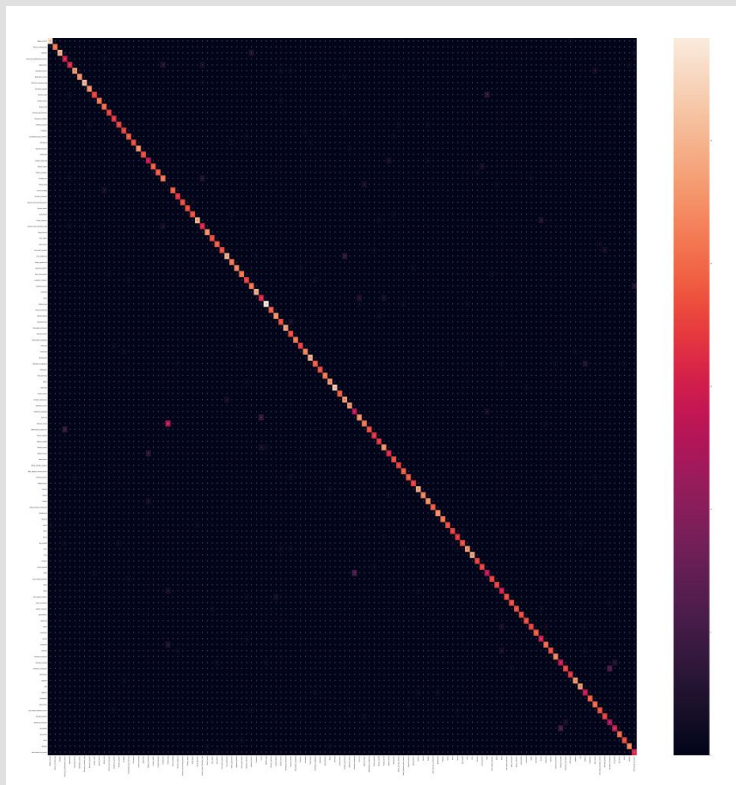
Pérdida



NASNet - Accuracy por categoría



NASNet - Matriz de confusión



Modelo Pre Entrenado Xception

Xception - Arquitectura

Capa	Activación
conv_base (xception)	
Flatten	
Dense (256)	relu
Dense (120)	softmax

Xception - Compilación

loss	categorical_crossentropy
optimizer	RAdam <ul style="list-style-type: none">• total_steps = 5000• warmup_proportion = 0.1• min_lr = 1e-7
metrics	categorical_accuracy

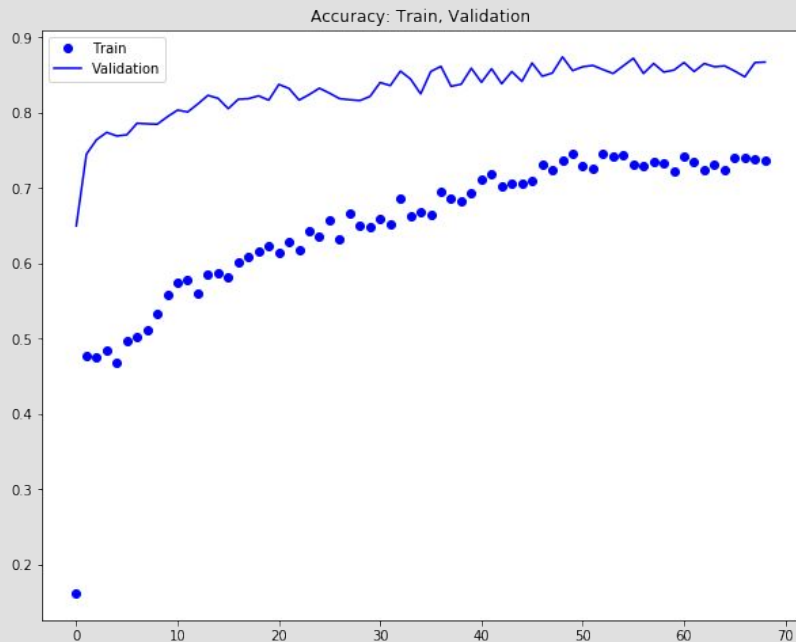
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
--	---

Xception - Entrenamiento

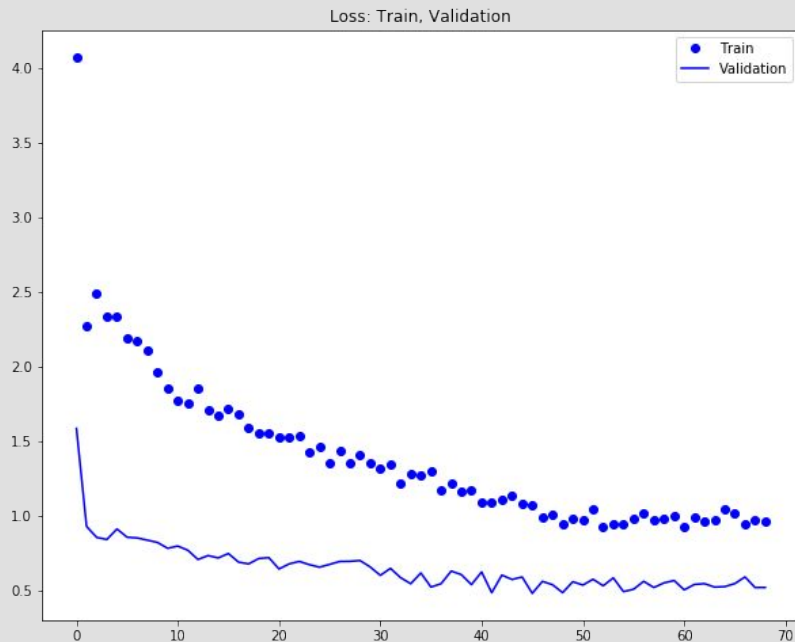
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 500• validation_data = valid_generator• validation_steps = 50

Xception - Resultados

Precisión



Pérdida



Modelo Pre Entrenado VGG19 / Fine Tunning

VGG19 / Fine Tunning - Arquitectura

Capa	Activación
conv_base (xception)	
Flatten	
Dense (256)	relu
Dense (120)	softmax

VGG19 / Fine Tunning - Compilación

loss	categorical_crossentropy
optimizer	<p>RAdam</p> <ul style="list-style-type: none">• total_steps = 5000• warmup_proportion = 0.1• min_lr = 1e-7
metrics	categorical_accuracy

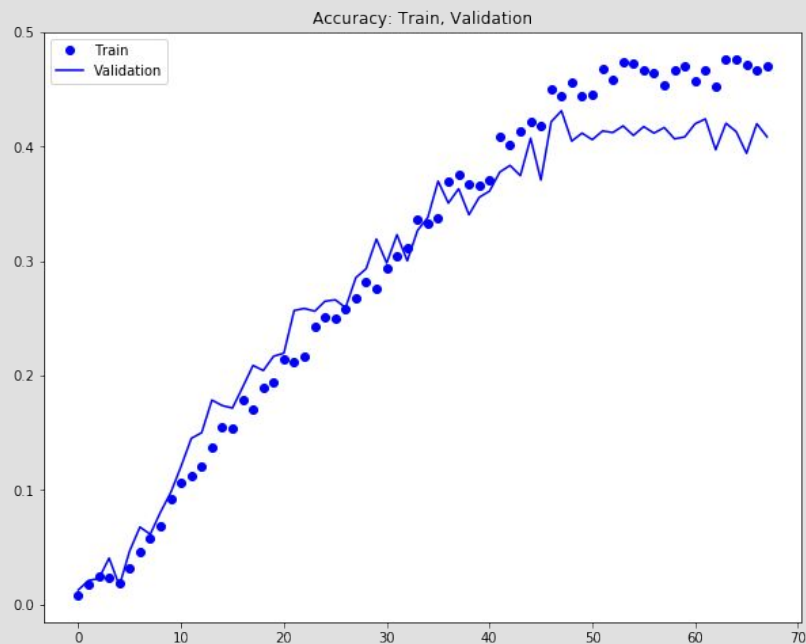
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

VGG19 / Fine Tuning - Entrenamiento

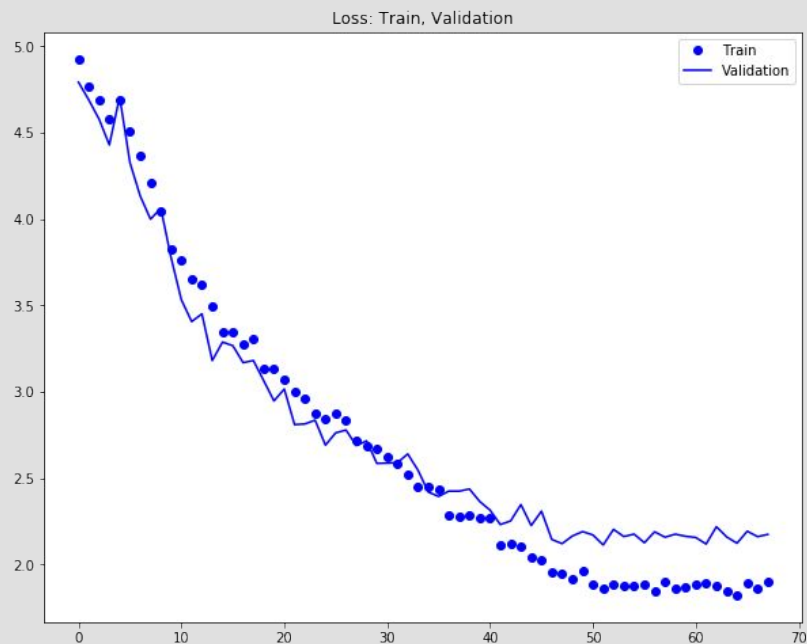
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 2000• validation_data = valid_generator• validation_steps = 50

VGG19 / Fine Tuning - Resultados

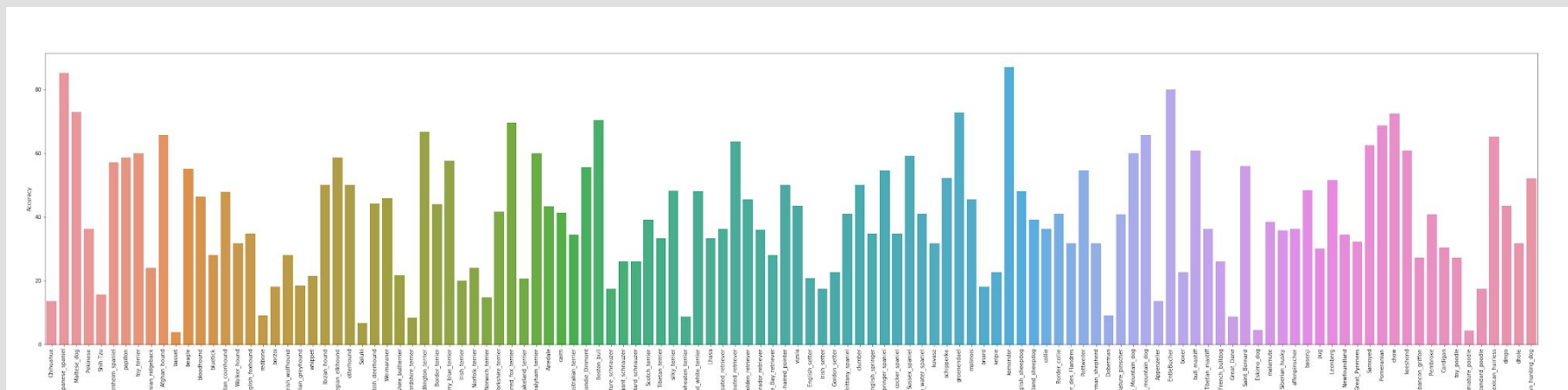
Precisión



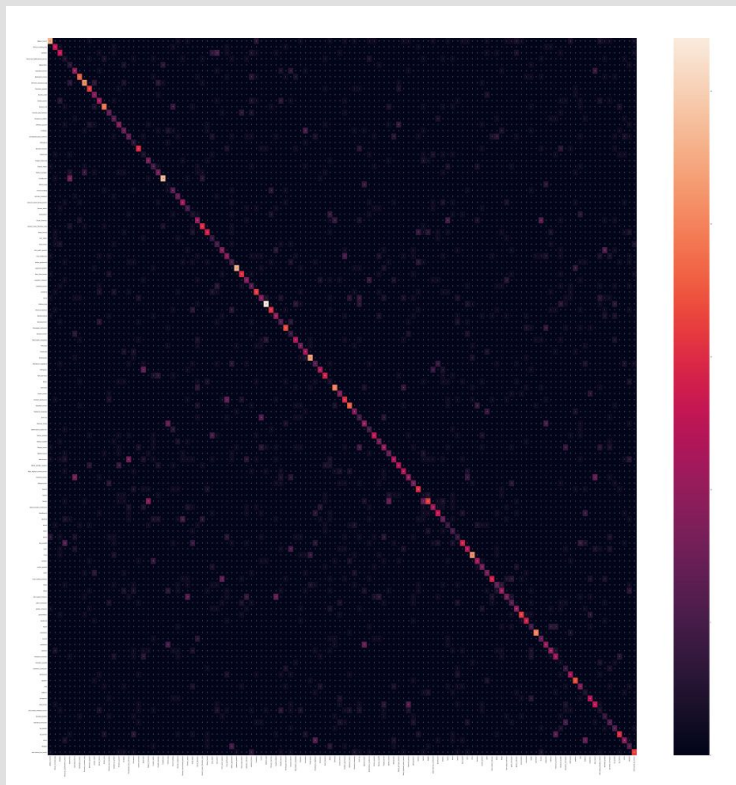
Pérdida



VGG19 / Fine Tuning - Accuracy por categoría



VGG19 / Fine Tunning - Matriz de confusión



Modelo Pre Entrenado InceptionV3 / Fine Tunning

InceptionV3 / Fine Tunning - Arquitectura

Capa	Activación
conv_base (xception)	
Flatten	
Dense (512)	relu
Dropout (0.1)	
Dense (256)	relu
Dropout (0.5)	
Dense (120)	softmax

InceptionV3 / Fine Tuning - Compilación

loss	categorical_crossentropy
optimizer	<p>RAdam</p> <ul style="list-style-type: none">• total_steps = 5000• warmup_proportion = 0.1• min_lr = 1e-7
metrics	categorical_accuracy

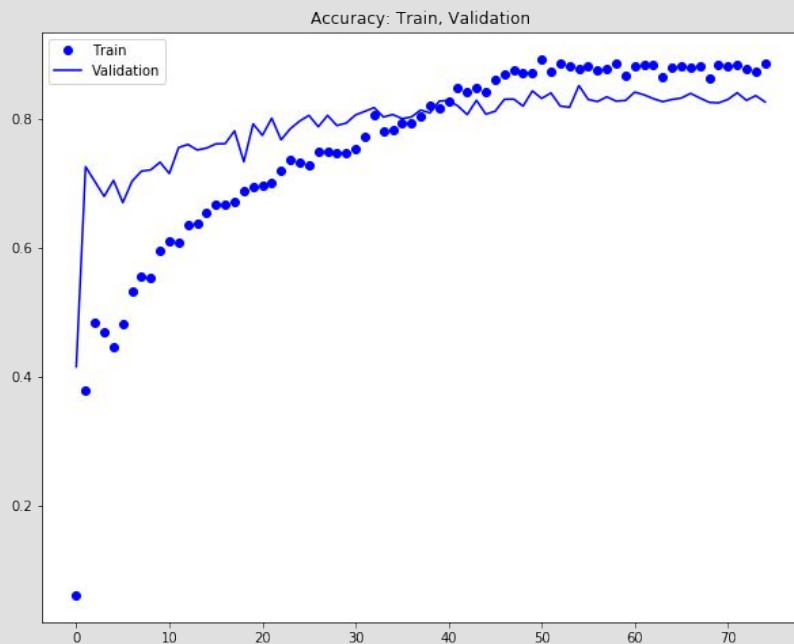
ImageDataGenerator (data augmentation)	<ul style="list-style-type: none">• rescale = 1./255• rotation_range = 40• width_shift_range = 0.2• height_shift_range = 0.2• shear_range = 0.2• zoom_range = 0.2• horizontal_flip = True
---	---

InceptionV3 / Fine Tuning - Entrenamiento

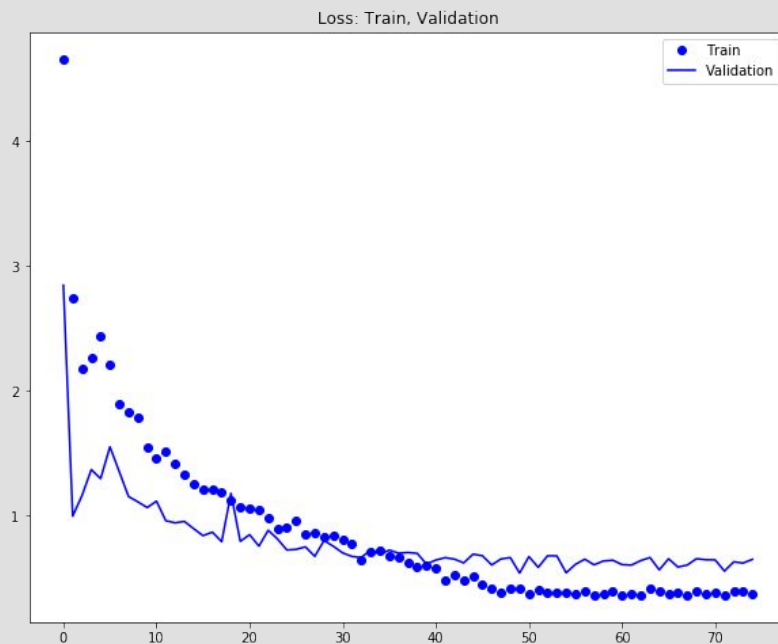
callbacks	<p>EarlyStopping</p> <ul style="list-style-type: none">• monitor = val_categorical_accuracy• min_delta = 0.001• patience = 20• restore_best_weights = True• mode = max
fit_generator	<ul style="list-style-type: none">• steps_per_epoch = 100• epochs = 2000• validation_data = valid_generator• validation_steps = 50

InceptionV3 / Fine Tuning - Resultados

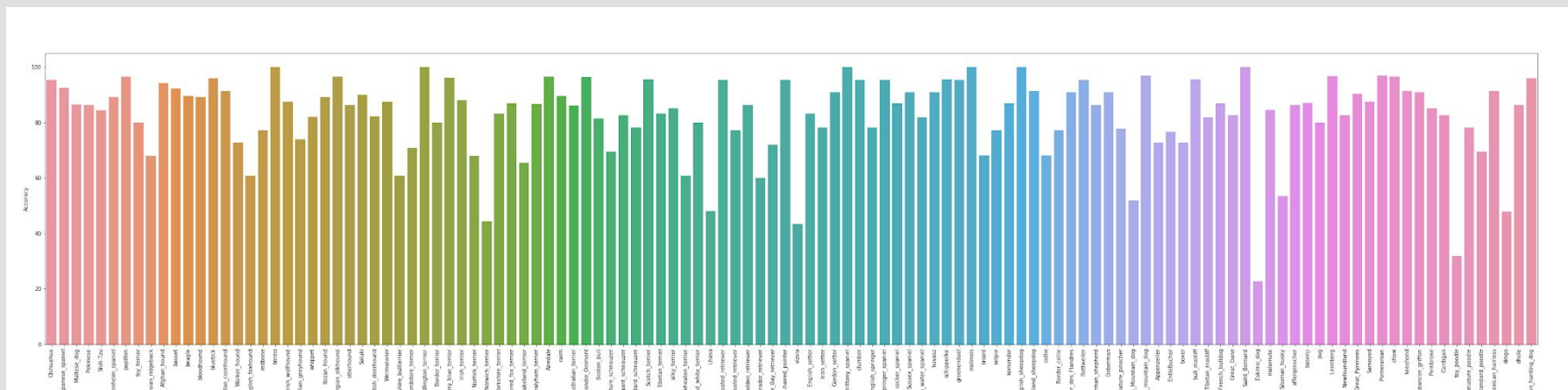
Precisión



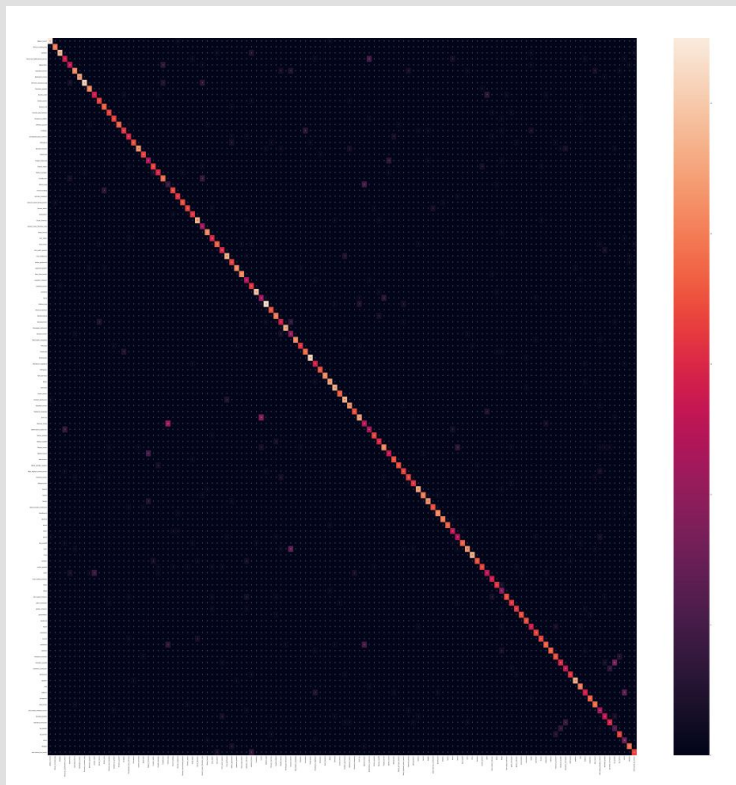
Pérdida



InceptionV3 / Fine Tuning - Accuracy por categoría



InceptionV3 / Fine Tuning - Matriz de confusión

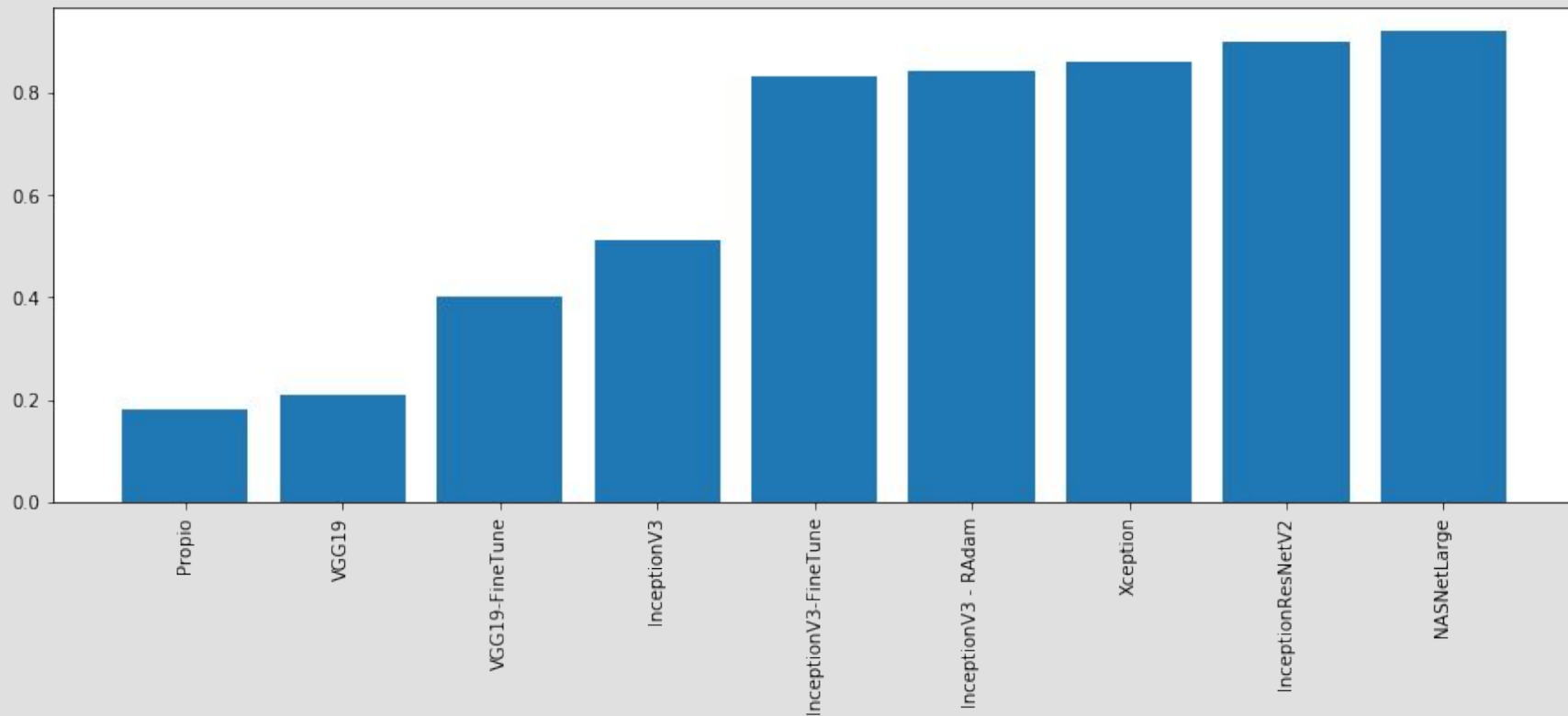


Conclusión

General Accuracy

NASNetLarge	0.92
InceptionResNetV2	0.90
Xception	0.86
InceptionV3 - RAdam	0.84
InceptionV3-FineTune	0.83
InceptionV3	0.51
VGG19-FineTune	0.40
VGG19	0.21
Propio	0.18

General Accuracy



Ejemplos de confusión

Ejemplo: Shih-Tzu = Lhasa

Shih-Tzu



Lhasa



Ejemplo: Shih-Tzu = Lhasa

Shih-Tzu



Lhasa



Ejemplo: Miniature poodle = Toy poodle

Miniature poodle



Toy poodle



Ejemplo: Miniature poodle = Toy poodle

Miniature poodle



Toy poodle



Ejemplo: Wire-haired fox terrier = Irish terrier

Wire-haired fox terrier



Irish terrier



Ejemplo: Wire-haired fox terrier = Irish terrier

Wire-haired fox terrier



Irish terrier



¿Preguntas?

Muchas gracias.