Tabla 1

Precisión

tor ->

flip

ImageDataGenera ImageDataGenera ImageDataGenera ImageDataGenera Capas

tor ->

brightness_range zoom_range

tor -> horizontal tor -> validation tor ->

Model.fit()

Patience

0,9565	n_range flip 10 True	6lip 0.2	brightness_range (0.8, 1.2)	oom_range 0,2	<pre>model = keras.Sequential() model.add(Rescaling(scale=(1./127.5),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es]</pre>	10
					<pre>model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>		
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9565	10 True	0.2	(0.8, 1.2)	-	<pre>optimizer=tf.keras.optimizers.Adam(1e-3), metrics=['accuracy'])</pre>	<pre>= model.fit(train_ds, epochs=epochs,</pre>	20
					<pre>input_shape=(150, 150, 3))) model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2)))</pre>	<pre>validation_data=val_ds, callbacks = [es]</pre>	
					<pre>model.add(Dropout(0.25)) model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9565	10 True	0.2	(0.6, 1.5)	-	<pre>optimizer=tf.keras.optimizers.Adam(1e-3), metrics=['accuracy'])</pre>	<pre>= model.fit(train_ds, epochs=epochs,</pre>	10
					<pre>input_shape=(150, 150, 3))) model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>	<pre>validation_data=val_ds, callbacks = [es]</pre>	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9441 10	10 True	0.2	-	-	<pre>model = keras.Sequential() model.add(Rescaling(scale=(1./127.5),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es]</pre>	20
					<pre>model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>		
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9379 0	0 True	0.2	(0.8, 1.2)	-	<pre>model.add(Rescaling(scale=(1./127.5),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds,</pre>	10
					<pre>model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>	callbacks = [es]	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9379 10	10 True	0.2	(0.8, 1.2)	-	<pre>model.add(Rescaling(scale=(1./127.5),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation data=val ds,</pre>	10
					<pre>input_shape=(150, 150, 3))) model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>	<pre>validation_data=val_ds, callbacks = [es], use_multiprocessing=True, workers=2</pre>	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
	10 True	0.2	(0.3, 0.6)	-	<pre>optimizer=tf.keras.optimizers.Adam(1e-3),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds</pre>	10
					<pre>input_shape=(150, 150, 3))) model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>	<pre>validation_data=val_ds, callbacks = [es]</pre>	
					<pre>model.add(Dropout(0.25)) model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9317	10 True	0.2	(0.8, 1.2)	-	<pre>metrics=['accuracy']) model = keras.Sequential() model.add(Rescaling(scale=(1./127.5),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds,</pre>	10
					<pre>model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax'))</pre>	callbacks = [es]	
0,9317 10	10 True	0.2	(0.8, 1.2)	-		= model.fit(10
					<pre>model.add(Rescaling(scale=(1./127.5),</pre>	<pre>train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es]</pre>	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9255 10	10 True	0.2	0.2 (0.8, 1.2)		<pre>model = keras.Sequential() model.add(Rescaling(scale=(1./127.5),</pre>	<pre>model.fit(train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es]</pre>	10
					<pre>model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dropout(0.5))</pre>		
	10 True	0.2	(0.8, 1.2)	-	<pre>model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>	= model.fit(10
					<pre>model.add(Rescaling(scale=(1./127.5),</pre>	<pre>train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es]</pre>	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9193	20 True	0.2	(0.8, 1.2)	-	metrics=['accuracy'])	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es]</pre>	10
					<pre>model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>	. []	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9130 10	10 True	0.2	(0.8, 1.2)	-	<pre>optimizer=tf.keras.optimizers.Adam(1e-3),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds,</pre>	10
					<pre>model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Flatten())</pre>	<pre>callbacks = [es], use_multiprocessing=True, workers=2, max_queue_size=2</pre>	
					<pre>model.add(Flatten()) model.add(Dense(64, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,9068 10	10 True	0.2	(0.8, 1.2)	-		<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es], use_multiprocessing=True,</pre>	10
					<pre>model.add(MaxPooling2D(pool_size=(2, 2))) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Flatten())</pre>	workers=2, max_queue_size=2	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,8696 10	10 True	0.2	(0.3, 0.6)	-		<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_ds, callbacks = [es]</pre>	10
					<pre>model.add(Conv2D(32, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(64, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25))</pre>		
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(9, activation='softmax')) model.compile(loss=tf.keras.losses.categorical_crossentropy,</pre>		
0,8571	10 True	0.2	(0.8, 1.2)	-	<pre>metrics=['accuracy']) model = keras.Sequential() model.add(Rescaling(scale=(1./127.5),</pre>	<pre>= model.fit(train_ds, epochs=epochs, validation_data=val_da</pre>	10
					<pre>input_shape=(150, 150, 3))) model.add(Conv2D(32, kernel_size=(3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Dropout(0.25)) model.add(Conv2D(128, (3, 3), activation='relu')) model.add(MaxPooling2D(pool_size=(2, 2)))</pre>	<pre>validation_data=val_ds, callbacks = [es]</pre>	
					<pre>model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5))</pre>		

0,8571

0,7888

0

10

10

10

0.2

0.2

0.2

True

True

True

(0.8, 1.2)

(0.3, 0.6)

(0.8, 1.2)

model.add(Dropout(0.5))

model = keras.Sequential()

model.add(Dropout(0.25))

model.add(Dropout(0.25))

model.add(Dropout(0.5))

model = keras.Sequential()

model.add(Dropout(0.25))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dropout(0.5))

model = keras.Sequential()

model.add(Dropout(0.25))

model.add(Dropout(0.5))

model.add(Flatten())

model.add(Flatten())

model.add(Dense(9, activation='softmax'))

model.add(Rescaling(scale=(1./127.5),

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Dense(64, activation='relu'))

model.add(Rescaling(scale=(1./127.5),

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Dense(128, activation='relu'))

model.add(Dense(9, activation='softmax'))

model.add(Rescaling(scale=(1./127.5),

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Dense(128, activation='relu'))

model.add(Dense(9, activation='softmax'))

model.add(Dense(9, activation='softmax'))

metrics=['accuracy'])

offset=-1,

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(Conv2D(128, (3, 3), activation='relu'))

metrics=['accuracy'])

offset=-1,

model.add(Conv2D(32, (3, 3), activation='relu'))

model.add(Conv2D(128, (3, 3), activation='relu'))

metrics=['accuracy'])

offset=-1,

model.add(Conv2D(128, (3, 3), activation='relu'))

metrics=['accuracy'])

model.compile(loss=tf.keras.losses.categorical_crossentropy,

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu'))

model.compile(loss=tf.keras.losses.categorical_crossentropy,

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu'))

model.compile(loss=tf.keras.losses.categorical_crossentropy,

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))

model.compile(loss=tf.keras.losses.categorical_crossentropy,

optimizer=tf.keras.optimizers.Adam(1e-3),

optimizer=tf.keras.optimizers.Adam(1e-3),

input_shape=(150, 150, 3)))

optimizer=tf.keras.optimizers.Adam(1e-3),

input_shape=(150, 150, 3)))

optimizer=tf.keras.optimizers.Adam(1e-3),

input_shape=(150, 150, 3)))

h = model.fit(

h = model.fit(

h = model.fit(

train_ds,

epochs=epochs,

validation_data=val_ds,
callbacks = [es]

train_ds,

epochs=epochs,

callbacks = [es]

validation_data=val_ds,

train_ds,

epochs=epochs,

callbacks = [es]

validation_data=val_ds,

10

10

10