

# مستندات پروژه چهارم هوش محاسباتی

اعضای گروه: الناز محمدی، زهرا رستمی

## فاز اول: Model Designing

ابتدا با Resize تصاویر و Normalization ، Preprocess انجام می‌دهیم. سپس 4 مدل شبکه عصبی را طراحی

کرده و دقت هریک را محاسبه کردیم:

```
1 model = Sequential([
2     Conv2D(64, (3, 3), activation='relu', input_shape=(64, 64, 3)),
3     Conv2D(128, (3, 3), activation='relu'),
4     MaxPooling2D((2, 2)),
5     Conv2D(256, (3, 3), activation='relu'),
6     MaxPooling2D((2, 2)),
7     GlobalAveragePooling2D(),
8     Dense(1024, activation='relu'),
9     Dense(512, activation='relu'),
10    Dense(256, activation='relu'),
11    Dense(10, activation='softmax')
12 ])
```

573/573 ————— 24s 42ms/step - accuracy: 0.9046 - loss: 0.2872  
Train Accuracy: 0.90  
164/164 ————— 9s 57ms/step - accuracy: 0.7072 - loss: 1.0371  
Test Accuracy: 0.70

```
1 model = Sequential([
2     Conv2D(64, (3, 3), activation='relu', input_shape=(64, 64, 3)),
3     MaxPooling2D((2, 2)),
4     Conv2D(128, (3, 3), activation='relu'),
5     MaxPooling2D((2, 2)),
6     Conv2D(256, (3, 3), activation='relu'),
7     GlobalAveragePooling2D(),
8     Dense(1024, activation='relu'),
9     Dense(512, activation='relu'),
10    Dense(256, activation='relu'),
11    Dense(10, activation='softmax')
12 ])
```

573/573 ————— 23s 40ms/step - accuracy: 0.8824 - loss: 0.3540  
Train Accuracy: 0.88  
164/164 ————— 9s 53ms/step - accuracy: 0.7105 - loss: 0.9796  
Test Accuracy: 0.70

```
1 model = Sequential([
2     Conv2D(64, (3, 3), activation='relu', input_shape=(64, 64, 3)),
3     Conv2D(128, (3, 3), activation='relu'),
4     Conv2D(256, (3, 3), activation='relu'),
5     MaxPooling2D((2, 2)),
6     MaxPooling2D((2, 2)),
7     GlobalAveragePooling2D(),
8     Dense(1024, activation='relu'),
9     Dense(512, activation='relu'),
10    Dense(256, activation='relu'),
11    Dense(10, activation='softmax')
12 ])
```

573/573 ————— 27s 48ms/step - accuracy: 0.8316 - loss: 0.4966  
Train Accuracy: 0.83  
164/164 ————— 12s 72ms/step - accuracy: 0.7055 - loss: 0.9100  
Test Accuracy: 0.70

```
1 model = Sequential([
2     Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
3     MaxPooling2D((2, 2)),
4     Conv2D(64, (3, 3), activation='relu'),
5     MaxPooling2D((2, 2)),
6     Conv2D(128, (3, 3), activation='relu'),
7     MaxPooling2D((2, 2)),
8     Flatten(),
9     Dense(256, activation='relu'),
10    Dense(128, activation='relu'),
11    Dense(10, activation='softmax')
12 ])
```

573/573 ————— 22s 39ms/step - accuracy: 0.9737 - loss: 0.0791  
Train Accuracy: 0.97  
164/164 ————— 9s 58ms/step - accuracy: 0.6133 - loss: 2.6133  
Test Accuracy: 0.63

با توجه به نتایج به دست آمده مدل 2 را به عنوان بهترین مدل با 1,292,682 پارامتر انتخاب کرده و

hyperparameter tuning انجام می‌دهیم:

batch\_size=16 , learning\_rate=0.0005 Train Accuracy: 0.90 , Test Accuracy: 0.72

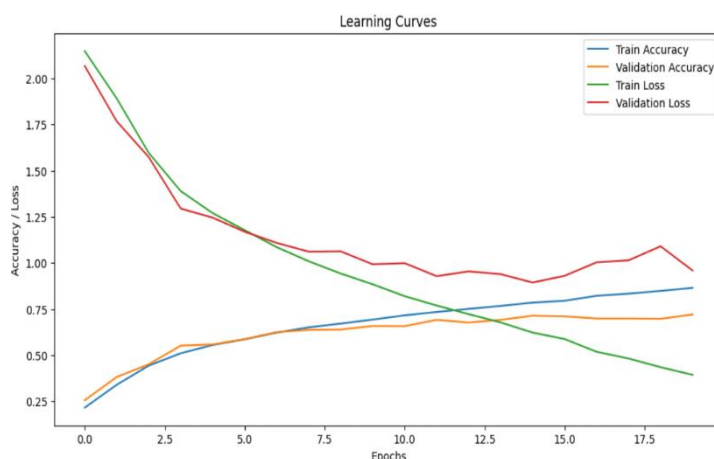
batch\_size=32 , learning\_rate=0.001 Train Accuracy: 0.90 , Test Accuracy: 0.70

batch\_size=64 , learning\_rate=0.0001 Train Accuracy: 0.54 , Test Accuracy: 0.52

batch\_size=128 , learning\_rate=0.01 Train Accuracy: 0.18 , Test Accuracy: 0.18

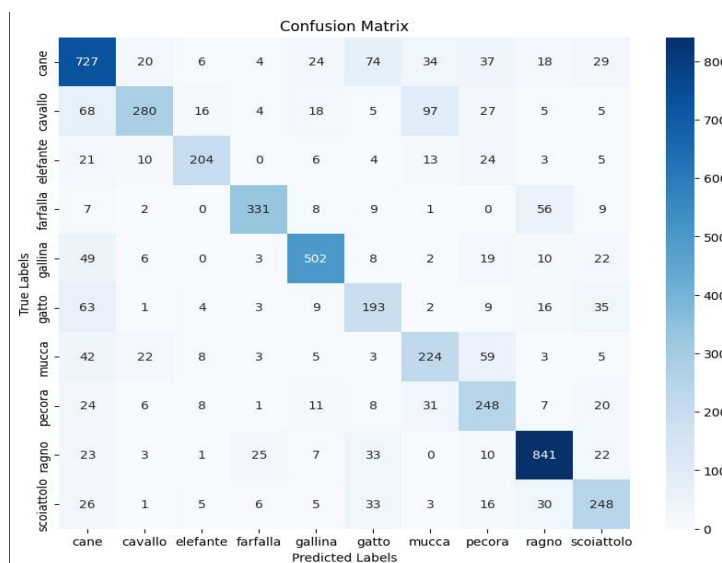
best -> batch\_size=16 , learning\_rate=0.0005

Train Loss به طور پیوسته کاهش داشته است. Validation Loss در ابتدا کاهش و بعد از حدود epoch 11 افزایش داشته که نشان دهنده Overfit است. Train Accuracy به صورت تدریجی افزایش پیدا کرده است و به بیش تر از 70٪ رسیده است. Validation Accuracy بعد از حدود epoch 11 تقریباً ثابت مانده که بازهم نشان دهنده Overfit است.



مدل برای برخی کلاس‌ها مانند "cane" و "ragno" عملکرد بسیار خوبی دارد و تعداد پیش‌بینی‌های درست بیشتر است.

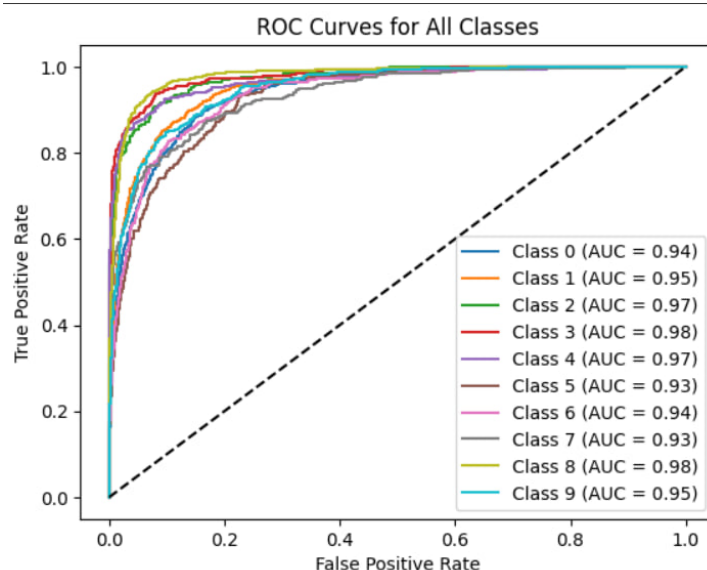
اما کلاس‌هایی مثل "mucca", "cavallo" دچار اشتباهات زیادی هستند. این می‌تواند ناشی از شباهت ظاهری یا ویژگی‌های کم‌تمایز در تصاویر این کلاس‌ها باشد.



AUC برای همه کلاس‌ها بالای 0.9 است، که نشان می‌دهد مدل قدرت تفکیک نسبی خوبی دارد.

کلاس‌هایی مانند "ragno" (8) و "farfalla" (3) عملکرد بهتری دارند (AUC نزدیک به 0.98)

کلاس‌هایی مانند "gatto" (5) و "pecora" (6) مقدار AUC کمتری نسبت به بقیه کلاس‌ها دارند (حدود 0.93)



همچنین نتایج ارزیابی مدل به صورت زیر است:

دقت کلی مدل برابر با 72٪ است.

کلاس‌هایی مثل "ragno" و "farfalla" و "gallina" بهترین عملکرد را

دارند (F1-Score بالای 0.8)

کلاس‌هایی مثل "gatto" و "pecora" و "mucca" عملکرد ضعیف‌تری

دارند (F1-Score حدود 0.55 تا 0.61)

	precision	recall	f1-score	support
cane	0.69	0.75	0.72	973
cavallo	0.80	0.53	0.64	525
elefante	0.81	0.70	0.75	290
farfalla	0.87	0.78	0.82	423
gallina	0.84	0.81	0.83	621
gatto	0.52	0.58	0.55	335
mucca	0.55	0.60	0.57	374
pecora	0.55	0.68	0.61	364
ragno	0.85	0.87	0.86	965
scoiattolo	0.62	0.66	0.64	373
accuracy			0.72	5243
macro avg	0.71	0.70	0.70	5243
weighted avg	0.73	0.72	0.73	5243

در ادامه 5 تا از تصاویر تست را به همراه لیبل واقعی و لیبل پیشبینی شده آن ویزوالایز کردیم:

True: 8, Pred: 8



True: 7, Pred: 7



True: 4, Pred: 9



True: 0, Pred: 0



True: 8, Pred: 8





## فاز دوم: Model Designing

در این فاز نیز مانند فاز قبلی ابتدا با Resize تصاویر و Normalization ، Preprocess انجام می‌دهیم. سپس مدل فاز قبلی را با دستکاری لایه fully connected به 392,634 پارامتر می‌رسانیم و hyperparameter tuning انجام می‌دهیم:

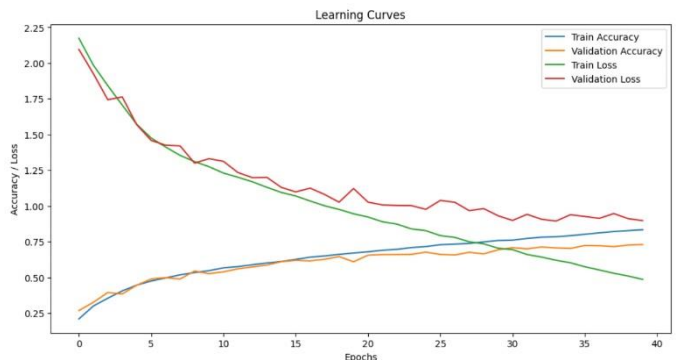
```
batch_size=16, learning_rate=0.0005, epochs=40 Train Accuracy: 0.87 , Test Accuracy: 0.73
batch_size=32, learning_rate=0.001, epochs=40 Train Accuracy: 0.89 , Test Accuracy: 0.71
batch_size=64, learning_rate=0.0001, epochs=40 Train Accuracy: 51 , Test Accuracy: 50
batch_size=128, learning_rate=0.01, epochs=40 Train Accuracy: 0.19 , Test Accuracy: 0.19
best -> batch_size=16, learning_rate=0.0005, epochs=40
```

```
1 model_2 = Sequential([
2     Conv2D(64, (3, 3), activation='relu', input_shape=(64, 64, 3)),
3     Conv2D(128, (3, 3), activation='relu'),
4     MaxPooling2D((2, 2)),
5     Conv2D(256, (3, 3), activation='relu'),
6     MaxPooling2D((2, 2)),
7     GlobalAveragePooling2D(),
8     Dense(64, activation='relu'),
9     Dense(64, activation='relu'),
10    Dense(16, activation='relu'),
11    Dense(10, activation='softmax')
12 ])
```

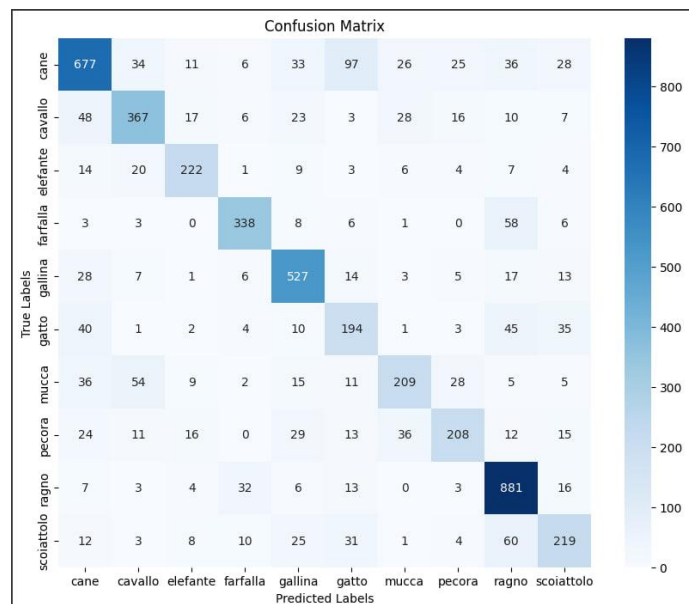
```
1146/1146 ————— 29s 25ms/step - accuracy: 0.8634 - loss: 0.4128
Train Accuracy: 0.87
328/328 ————— 8s 26ms/step - accuracy: 0.7170 - loss: 0.9078
Test Accuracy: 0.73
```

سپس مانند فاز قبل نمودارهای یادگیری و ماتریس گمراهی و ROC\_Curve را رسم می‌کنیم:

Train Loss به طور پیوسته کاهش داشته است.  
Validation Loss کاهش یافته و نوساناتی را در طول این کاهش مشاهده می‌کنیم. Train Accuracy به تدریج افزایش یافته و به حدود 90٪ رسیده است.  
Validation Accuracy ابتدا افزایش یافته، و در حدود 70-75٪ ثابت می‌ماند.



مدل برای برخی کلاس‌ها مانند "cane" و "ragno" عملکرد بسیار خوبی دارد و تعداد پیش‌بینی‌های درست بیشتر است. و کلاسی مثل "gatto" پیش‌بینی‌های درست کم‌تری دارد.  
طبقه‌بندی‌های نادرست برای "gatto", "scoiattolo" و "mucca" بالا است، که نشان‌دهنده عدم تعادل احتمالی کلاس یا شباهت‌های بین این کلاس‌ها است.

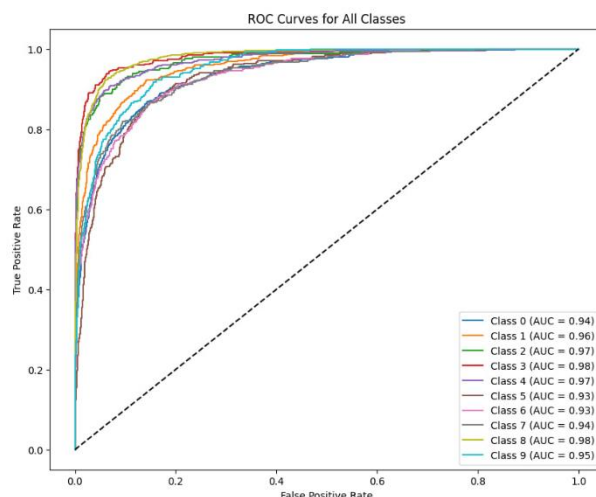


AUC برای همه کلاس‌ها بالای 0.9 است، که نشان دهنده عملکرد خوب است.

کلاس‌هایی مانند "ragno" (8) و "farfalla" (3) عملکرد بهتری دارند (AUC نزدیک به 0.98)

کلاس‌هایی مانند "gatto" (5) و "pecora" (6)، AUC کمتری نسبت به بقیه کلاس‌ها دارند (حدود 0.93)

مشاهده می‌شود نتایج بسیار نزدیک به فاز اول می‌باشد.



همچنین نتایج ارزیابی مدل به صورت زیر است:

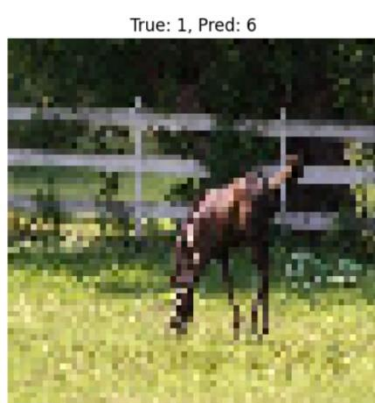
دقت کلی مدل برابر با 73٪ است.

کلاس‌هایی مثل "ragno" و "farfalla" و "gallina" بهترین عملکرد را دارند (F1-Score بالای 0.8)

کلاس‌هایی مثل "gatto" و "scoiattolo" و "mucca" عملکرد ضعیف‌تری دارند (F1-Score حدود 0.54 تا 0.61)

	precision	recall	f1-score	support
cane	0.76	0.70	0.73	973
cavallo	0.73	0.70	0.71	525
elefante	0.77	0.77	0.77	290
farfalla	0.83	0.80	0.82	423
gallina	0.77	0.85	0.81	621
gatto	0.50	0.58	0.54	335
mucca	0.67	0.56	0.61	374
pecora	0.70	0.57	0.63	364
ragno	0.78	0.91	0.84	965
scoiattolo	0.63	0.59	0.61	373
accuracy			0.73	5243
macro avg	0.71	0.70	0.71	5243
weighted avg	0.73	0.73	0.73	5243

در ادامه 5 تا از تصاویر تست را به همراه لیبل واقعی و لیبل پیشبینی شده آن ویزوالایز کردیم:



## فاز سوم: Overfitting Prevent

مدل اول:

با کمک Data Augmentation ( تغییر روشنایی، چرخش تصویر، جابجایی عرضی و ارتفاعی، بزرگنمایی یا کوچک‌نمایی تصویر، آینه‌ای کردن افقی تصویر) سعی شده از overfit جلوگیری شود. همچنین از مواردی مثل Weight Regularization و Dropout و Normalization Layers نیز برای کاهش overfit استفاده شده است. همچنین hyperparameter tuning انجام می‌دهیم.

batch\_size=16 , learning\_rate=0.00008 , epochs=60 , optimizer=RMSprop Train Accuracy: 0.77 , Test Accuracy: 0.72

batch\_size=16 , learning\_rate=0.00008 , epochs=60 , optimizer=adam Train Accuracy: 0.82 , Test Accuracy: 0.76

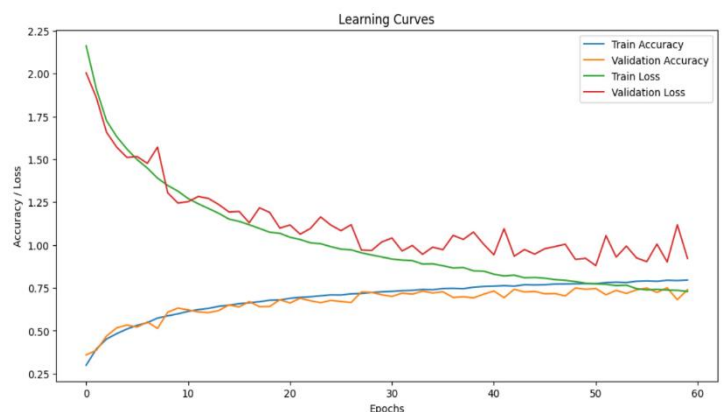
batch\_size=16 , learning\_rate=0.00008 , epochs=60 , optimizer=adam Train Accuracy: 0.79 , Test Accuracy: 0.75

best -> batch\_size=16 , learning\_rate=0.00008 , epochs=60

سپس مانند فاز قبل نمودارهای یادگیری و ماتریس گمراهی و ROC\_Curve را رسم می‌کنیم:

مدل تا حد زیادی همگراست.

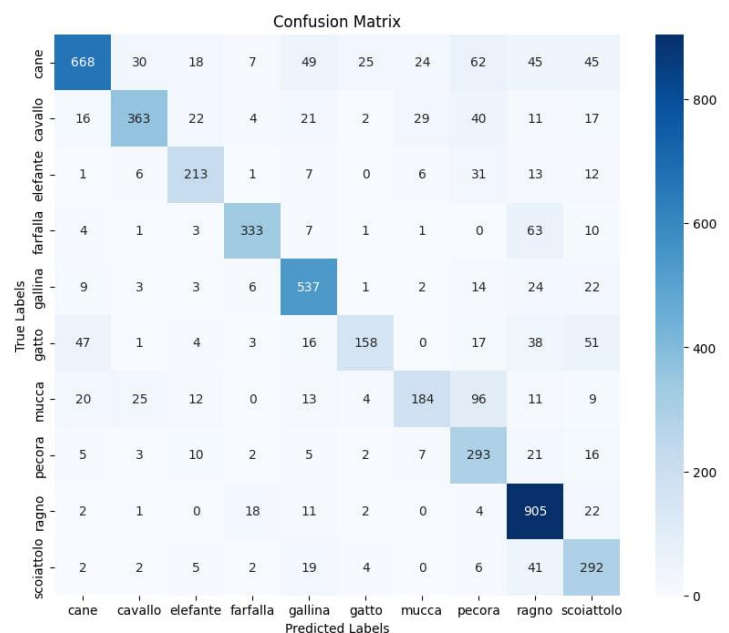
دقت داده‌های train و validation به مرور افزایش یافته و به حدود 0.75 رسیده است. خطاها به تدریج کاهش داشته‌اند، اما به دلیل نویز validation، احتمالاً مدل تا حدودی تحت تاثیر نوسانات قرار گرفته است.



کلاس‌های ragno و cane عملکرد بهتری دارند و تعداد پیش‌بینی‌های درستشان نسبت به کلاس‌های دیگر بیش‌تر است.

کلاس‌های gatto و mucca عملکرد ضعیف‌تری دارند و مدل بیشتر در این کلاس‌ها اشتباه می‌کند.

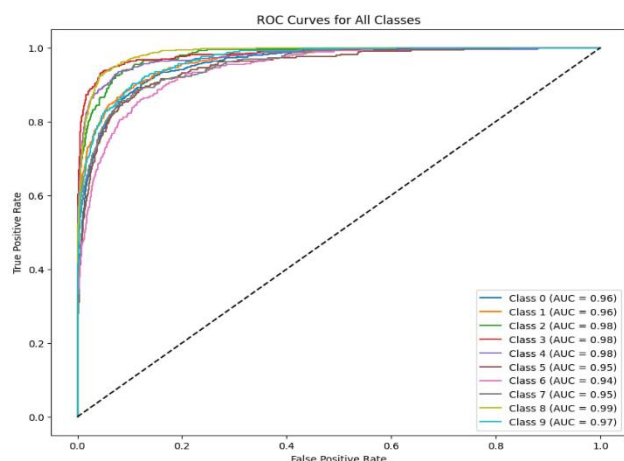
اشتباه مدل بین mucca (گاو) و pecora (گوسفند) به دلیل شباهت احتمالی ویژگی‌های این دو حیوان رخ می‌دهد.



مقدار AUC برای همه کلاس‌ها بالای 0.9 است که یعنی مدل توانایی خوبی در تفکیک کلاس‌ها از هم دارد.

کلاس ragno بالاترین مقدار AUC (0.99) را دارد و بهترین عملکرد را از بین کلاس‌ها دارد.

کلاس mucca با AUC کمتر از همه (0.94)، عملکرد ضعیف‌تری دارد.



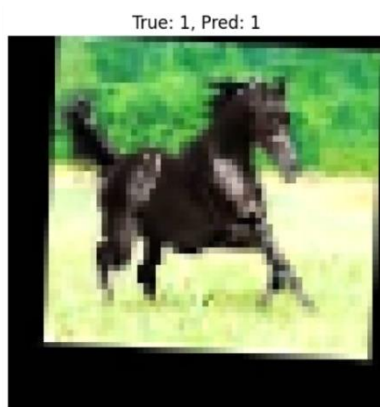
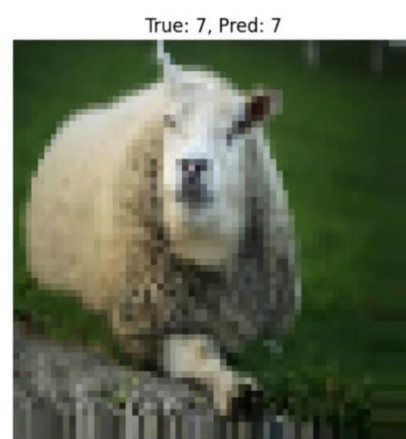
دقت کلی مدل برابر با 75٪ است.

کلاس‌هایی مثل "ragno" و "farfalla" و "gallina" بهترین عملکرد را دارند (F1-Score بالای 0.8)

کلاس‌هایی مثل "gatto" و "mucca" عملکرد ضعیف‌تری دارند (F1-Score 0.59)

	precision	recall	f1-score	support
cane	0.86	0.69	0.76	973
cavallo	0.83	0.69	0.76	525
elefante	0.73	0.73	0.73	290
farfalla	0.89	0.79	0.83	423
gallina	0.78	0.86	0.82	621
gatto	0.79	0.47	0.59	335
mucca	0.73	0.49	0.59	374
pecora	0.52	0.80	0.63	364
ragno	0.77	0.94	0.85	965
scoiattolo	0.59	0.78	0.67	373
accuracy			0.75	5243
macro avg	0.75	0.73	0.72	5243
weighted avg	0.77	0.75	0.75	5243

در ادامه 5 تا از تصاویر تست را به همراه لیبل واقعی و لیبل پیشبینی شده آن ویزوالایز کردیم:





## مدل دوم:

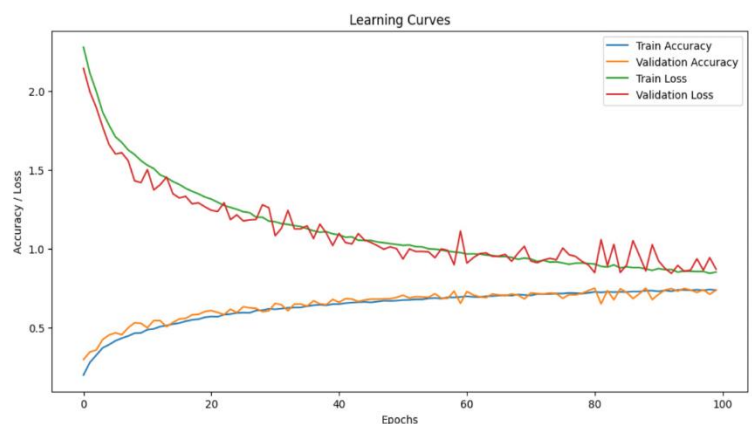
برای جلوگیری از overfit مانند مدل اول عمل می‌کنیم و از متدهای مشابه استفاده می‌کنیم. همچنین hyperparameter tuning انجام می‌دهیم.

```
batch_size=16 , learning_rate=0.00008 , epochs=60 , optimizer=RMSprop Train Accuracy: 0.71 , Test Accuracy: 0.68
batch_size=16 , learning_rate=0.00008 , epochs=60 , optimizer=adam Train Accuracy: 0.76 , Test Accuracy: 0.72
batch_size=16 , learning_rate=0.00009 , epochs=60 , optimizer=adam Train Accuracy: 76 , Test Accuracy: 71
batch_size=16 , learning_rate=0.00008 , epochs=100 , optimizer=adam Train Accuracy: 79 , Test Accuracy: 74
best -> batch_size=16 , learning_rate=0.00008 , epochs=100
```

مانند فازهای قبل نمودارهای یادگیری و ماتریس گمراهی و ROC\_Curve را رسم می‌کنیم:

مدل تا حد زیادی همگراست.

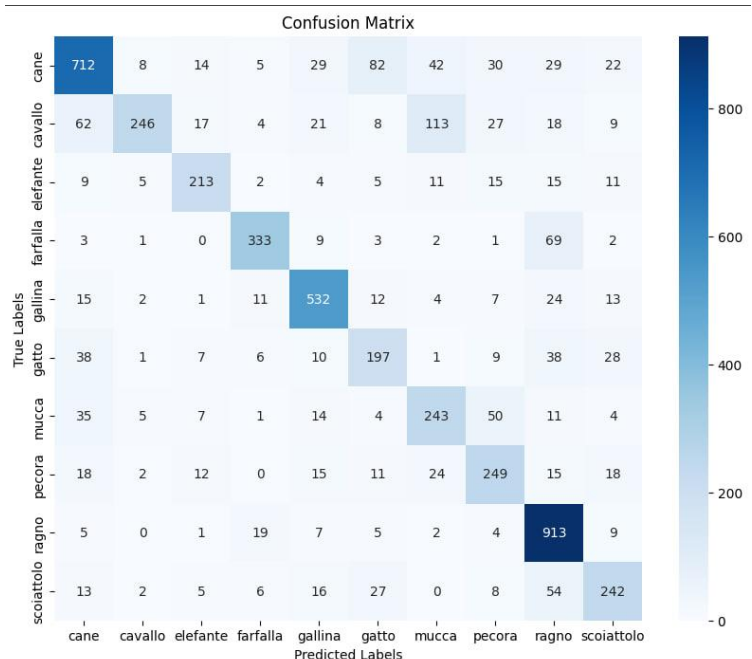
دقت داده‌های train و validation به مرور افزایش یافته اما نوسانات جزئی در validation دیده می‌شود. خطاها به تدریج کاهش داشته‌اند، اما به دلیل نویز validation، احتمالاً مدل تا حدودی تحت تاثیر نوسانات قرار گرفته است.



کلاس‌های cane و ragno عملکرد بهتری دارند و تعداد پیش‌بینی‌های درستشان نسبت به کلاس‌های دیگر بیش‌تر است.

کلاس‌های gatto و elefante عملکرد ضعیف‌تری دارند و مدل بیش‌تر در این کلاس‌ها اشتباه می‌کند.

اشتباه مدل بین mucca و cavallo همچنین بین gatto و cane نیز بیش‌تر از کلاس‌های دیگر است.

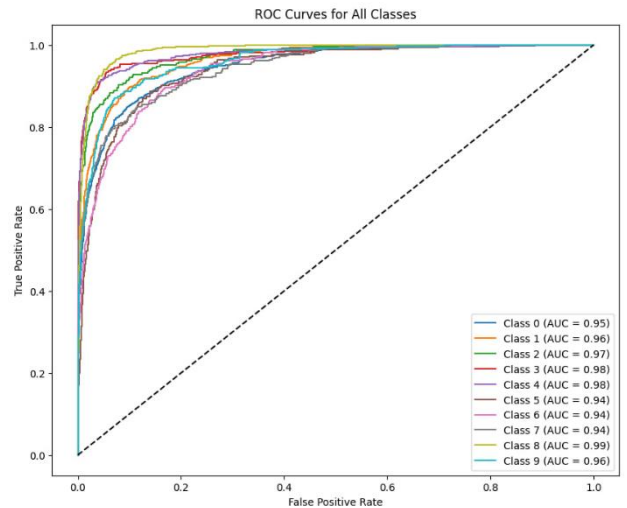




مقدار AUC برای همه کلاس‌ها بالای 0.9 است که یعنی مدل توانایی خوبی در تفکیک کلاس‌ها از هم دارد.

کلاس ragno بالاترین مقدار AUC (0.99) را دارد و بهترین عملکرد را از بین کلاس‌ها دارد.

کلاس‌های mucca و pecora و gatto با AUC کمتر از همه (0.94)، عملکرد ضعیف‌تری دارند.



دقت کلی مدل برابر با 74٪ است.

کلاس‌هایی مثل "ragno" و "farfalla" و "gallina" بهترین عملکرد را دارند (F1-Score بالای 0.8)

کلاس "gatto" عملکرد ضعیف‌تری دارند (F1-Score 0.57)

	precision	recall	f1-score	support
cane	0.78	0.73	0.76	973
cavallo	0.90	0.47	0.62	525
elefante	0.77	0.73	0.75	290
farfalla	0.86	0.79	0.82	423
gallina	0.81	0.86	0.83	621
gatto	0.56	0.59	0.57	335
mucca	0.55	0.65	0.60	374
pecora	0.62	0.68	0.65	364
ragno	0.77	0.95	0.85	965
scoiattolo	0.68	0.65	0.66	373
accuracy			0.74	5243
macro avg	0.73	0.71	0.71	5243
weighted avg	0.75	0.74	0.74	5243

در ادامه 5 تا از تصاویر تست را به همراه لیبل واقعی و لیبل پیشبینی شده آن و ویژوالایز کردیم:

