

پروژه چهارم: شبکه عصبی پیچشی CNN

عطیه بنکدار ۹۹۲۲۷۶۲۴۱۳

فاز اول:

۱. پیش پردازش داده ها:
نحوه پیاده سازی:

تعریف مسیر و پارامترها:

سایز تصاویر به 128×128 تغییر داده می شود.

اندازه دسته های (batch) داده برای پردازش ۳۲ است.

اعمال تبدیلات روی تصاویر:

استفاده از transforms.Compose برای تغییر سایز تصاویر به (128x128) ، تبدیل تصاویر به فرمت تانسور PyTorch ،

نرمال سازی مقادیر پیکسل ها به بازه [-1,1]

با استفاده از DataLoader ، داده ها به صورت دسته ای بارگذاری می شوند و ترتیب داده ها برای مجموعه آموزشی تصادفی سازی می شود.

مزایا و دلایل استفاده از این روش:

انعطاف پذیری در پردازش تصاویر: تغییر سایز و نرمال سازی مقادیر به بهبود عملکرد مدل کمک می کند.

مدیریت حافظه: تقسیم داده ها به دسته های کوچکتر (batch) مانع از اشغال بیش از حد حافظه می شود.

تقسیم تصادفی داده ها: این روش به کاهش بایاس و ایجاد مجموعه داده های متعادل کمک می کند.

تحلیل خروجی ها:

تعداد نمونه ها در هر مجموعه چاپ شده است، که تأیید می کند تقسیم داده ها به درستی انجام شده است.

Train: 18325, Validation: 2617, Test: 5237

۲. تحلیل طراحی مدل CNN:

معماری مدل

لایه های کانولوشنی: لایه های کانولوشنی اولیه مدل مناسب برای استخراج ویژگی های ابتدایی از تصاویر هستند. فیلترهای ۳۲ و ۶۴ مناسب برای شناسایی ویژگی هایی مانند لبه ها، بافت ها و الگوهای کلی در تصاویر حیوانات می باشند.

با استفاده از پدینگ ۱ در لایه های کانولوشنی، ابعاد تصویر حفظ می شود، که به مدل این امکان را می دهد تا ویژگی ها را در ابعاد کامل استخراج کند.

Conv1: اولین لایه کانولوشن با ۳ کانال ورودی (RGB) و ۳۲ فیلتر، سایز هسته 3×3 و پدینگ ۱. خروجی این لایه یک تصویر با ابعاد $128 \times 128 \times 32$ است.

Conv2: دومین لایه کانولوشن با ۳۲ کانال ورودی و ۶۴ فیلتر، سایز هسته ۳x۳ و پدینگ ۱. این لایه خروجی ۶۴x۶۴x۶۴ تولید می‌کند.

لایه‌های Pooling: عملیات MaxPooling برای کاهش ابعاد و کاهش تعداد پارامترهای مدل انجام می‌شود که از لحاظ محاسباتی مفید است. این کاهش ابعاد کمک می‌کند تا مدل کمتر مستعد Overfitting شود.

MaxPool2d: از این لایه برای کاهش ابعاد و استخراج ویژگی‌های مهم استفاده می‌شود. سایز هسته ۲x۲ باعث می‌شود ابعاد هر تصویر به نصف کاهش یابد.

لایه‌های Fully Connected: لایه‌های Fully Connected با ۲۵۶ نورون به مدل کمک می‌کنند تا ویژگی‌های سطح بالاتری از داده‌ها استخراج کند. تعداد نورون‌ها متناسب با اندازه ورودی است.

لایه آخر (FC2) خروجی ۱۰ کلاسه برای شناسایی هر کدام از ۱۰ حیوان مختلف فراهم می‌آورد.

FC1: این لایه به تعداد ۲۵۶ نورون برای تبدیل ویژگی‌های استخراج شده به ویژگی‌های سطح بالاتر، از ورودی ابعاد ۳۲x۳۲x۶۴ (که معادل ۶۵۵۳۶ است) استفاده می‌کند.

FC2: در نهایت، لایه آخر یک لایه خطی است که خروجی ۱۰ بعدی (یک برای هر کلاس) را تولید می‌کند.

محاسبه تعداد پارامترها:

$$\text{Conv1: تعداد پارامترها} = 32 + (32 * 3 * 3) = 896$$

$$\text{Conv2: تعداد پارامترها} = 64 + (64 * 3 * 3 * 32) = 18496$$

$$\text{FC1: تعداد پارامترها} = 256 + (256 * 32 * 32 * 64) = 16776336$$

$$\text{FC2: تعداد پارامترها} = 10 + (10 * 256) = 2570$$

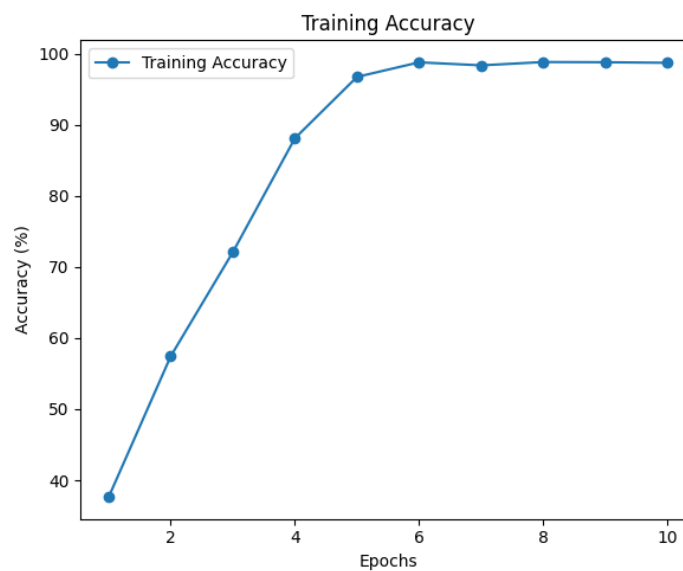
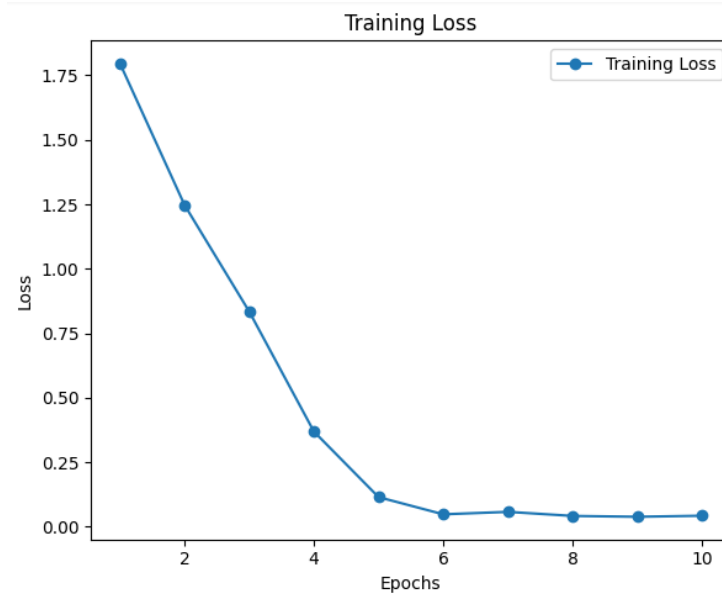
در مجموع، تعداد پارامترها برابر با ۱۶۷۸۹۹۴۲ است.

```
SimpleCNN(  
    (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (conv2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (fc1): Linear(in_features=65536, out_features=256, bias=True)  
    (fc2): Linear(in_features=256, out_features=10, bias=True)  
)
```

۳. تست و یادگیری:

مدل با نرخ یادگیری ۰/۰۰۱ و به مدت ۱۰ دوره (Epoch) آموزش داده شده است. کاهش یکنواخت مقدار خطا (Loss) نشان‌دهنده همگرایی خوب مدل است. دقت مدل از ۳۷.۶۳٪ در دوره اول به ۹۸.۷۱٪ در دوره دهم افزایش یافته است. کاهش قابل توجه خطا از ۱.۷۹۶۸ به ۰.۰۴۲۵.

```
Epoch 1/10, Loss: 1.7968, Accuracy: 37.63%  
Epoch 2/10, Loss: 1.2472, Accuracy: 57.44%  
Epoch 3/10, Loss: 0.8340, Accuracy: 72.08%  
Epoch 4/10, Loss: 0.3688, Accuracy: 88.10%  
Epoch 5/10, Loss: 0.1146, Accuracy: 96.73%  
Epoch 6/10, Loss: 0.0480, Accuracy: 98.77%  
Epoch 7/10, Loss: 0.0575, Accuracy: 98.35%  
Epoch 8/10, Loss: 0.0417, Accuracy: 98.82%  
Epoch 9/10, Loss: 0.0384, Accuracy: 98.79%  
Epoch 10/10, Loss: 0.0425, Accuracy: 98.71%
```



۴. ارزیابی مدل:

مدل روی مجموعه تست به دقت ۵۷.۴۴٪ رسیده است، این اختلاف نشان‌دهنده مشکل احتمالی بیش‌برازش (Overfitting) است. دقت بسیار بالای مدل در آموزش و کاهش خطای قابل توجه نشان‌دهنده تطبیق بیش‌ازحد با داده‌های آموزش است. همچنین احتمال دارد مجموعه تست شامل تصاویر با توزیع متفاوت یا ویژگی‌هایی باشد که مدل در داده‌های آموزشی به اندازه کافی ندیده است.

Test Accuracy: 57.44%

۵. تحلیل معیارهای ارزیابی مدل

Precision: درصد پیش‌بینی‌های درست برای هر کلاس از تمام پیش‌بینی‌های انجام شده در آن کلاس.

به طور متوسط، دقت مدل ۵۶٪ است. کلاس‌هایی مانند ragno ۷۷٪ و farfalla ۷۵٪ دقت بالاتری دارند، در حالی که gatto ۳۸٪ دقت پایینی دارد.

Recall: درصد نمونه‌های درست شناسایی‌شده برای هر کلاس از کل نمونه‌های واقعی آن کلاس. این معیار برای کلاس‌هایی مانند cane ۶۳٪ بالاتر است، اما برای pecora ۳۸٪ پایین‌تر است.

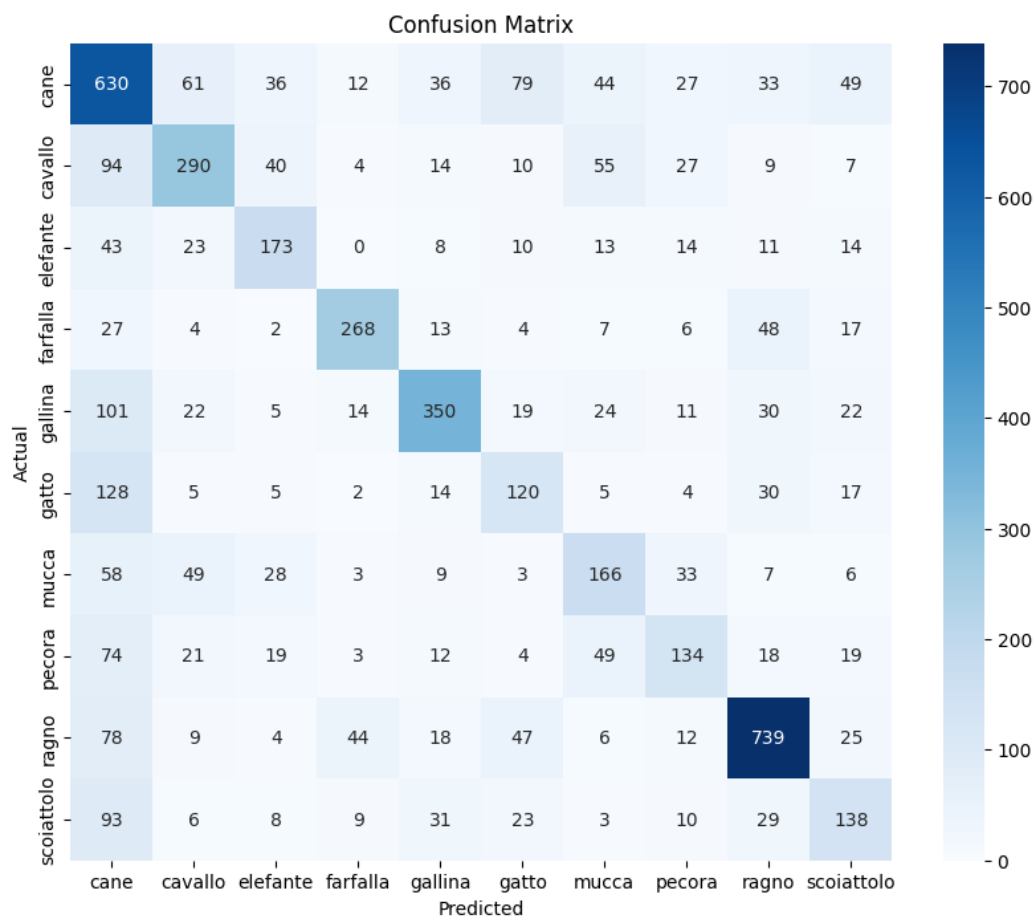
F1-Score: میانگین هارمونیک دقت و بازخوانی که عملکرد کلی مدل را برای هر کلاس نشان می‌دهد.

ragno با F1-Score برابر ۷۶٪ بهترین عملکرد را دارد، اما gatto با ۳۷٪ عملکرد ضعیفی دارد.

دقت کل (Accuracy): مدل توانسته به دقت ۵۷٪ دست یابد، که پایین‌تر از حد مطلوب است.

Confusion Matrix: این ماتریس نشان‌دهنده توزیع پیش‌بینی‌های درست و اشتباه برای هر کلاس است. کلاس‌های cane و ragno پیش‌بینی‌های صحیح بیشتری دارند. کلاس‌هایی مانند gatto و gallina نمونه‌های زیادی را به اشتباه در دیگر کلاس‌ها دسته‌بندی کرده‌اند.

Classification Report:				
	precision	recall	f1-score	support
cane	0.48	0.63	0.54	1007
cavallo	0.59	0.53	0.56	550
elefante	0.54	0.56	0.55	309
farfalla	0.75	0.68	0.71	396
gallina	0.69	0.59	0.63	598
gatto	0.38	0.36	0.37	330
mucca	0.45	0.46	0.45	362
pecora	0.48	0.38	0.42	353
ragno	0.77	0.75	0.76	982
scoiattolo	0.44	0.39	0.42	350
accuracy			0.57	5237
macro avg	0.56	0.53	0.54	5237
weighted avg	0.58	0.57	0.58	5237



۶. تصویر سازی پیش بینی ها:

تصاویر غیر نرمال سازی میشوند تا واقعی تر بشوند سپس پیشبینی چند تصویر با لیبل واقعی آن بررسی خواهد شد. چند نمونه مثال:

Actual: pecora, Predicted: cane



Actual: scoiattolo, Predicted: gallina



Actual: elefante, Predicted: elefante



Actual: farfalla, Predicted: farfalla



۷. تحلیل ROC Curve : رسم منحنی ROC (Receiver Operating Characteristic) برای ارزیابی عملکرد مدل در شناسایی هر کلاس و محاسبه مقدار AUC (Area Under Curve) برای هر کلاس.

عملکرد:

محاسبه احتمالات: از خروجی شبکه، احتمالات پیش‌بینی برای هر کلاس استخراج می‌شود. (Softmax)

برچسب‌گذاری باینری: برچسب‌های واقعی به فرم باینری (One-vs-Rest) تبدیل می‌شوند.

محاسبه منحنی ROC : نرخ مثبت‌های واقعی (TPR) و نرخ مثبت‌های کاذب (FPR) برای هر کلاس محاسبه می‌شود.

مساحت زیر منحنی (AUC) برای هر کلاس به عنوان معیاری از توانایی مدل در تمایز آن کلاس محاسبه می‌شود.

رسم نمودار: منحنی ROC برای هر کلاس به همراه مقدار AUC رسم می‌شود.

مزایا:

ارزیابی جزئی: هر کلاس به طور جداگانه بررسی می‌شود، که نقاط قوت و ضعف مدل را برای شناسایی هر کلاس مشخص می‌کند.

AUC به عنوان معیار کلی: مقدار AUC برای هر کلاس نشان‌دهنده کیفیت مدل در پیش‌بینی آن کلاس است:

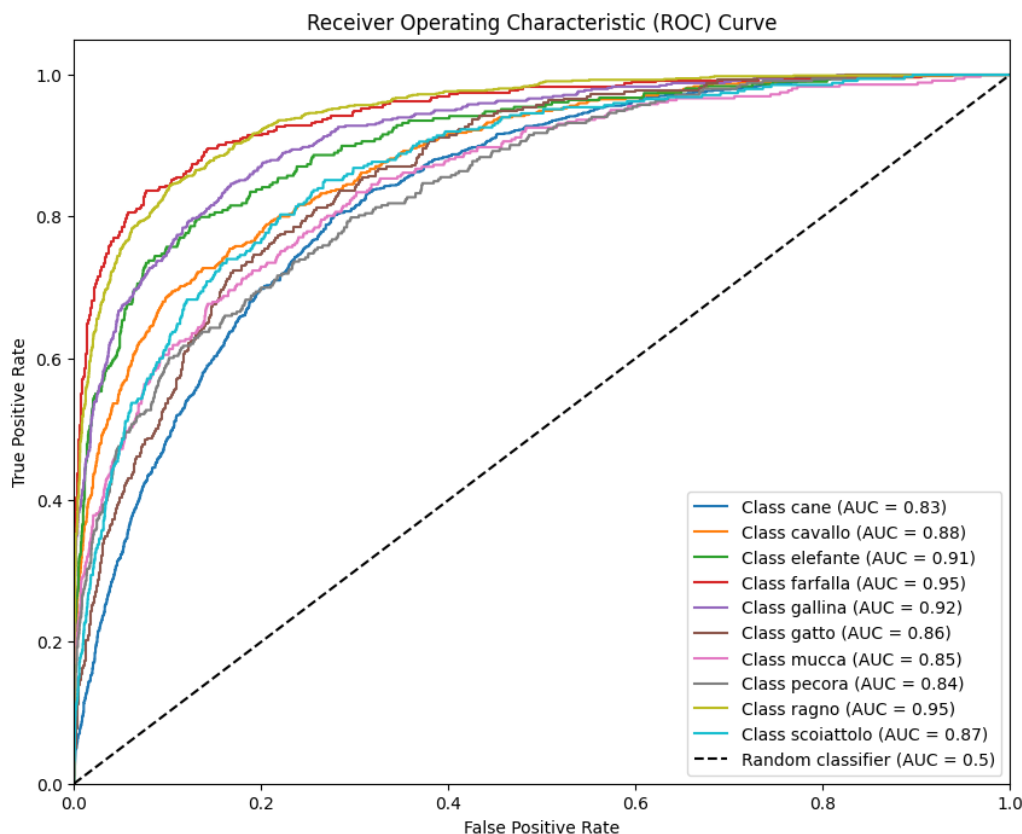
AUC نزدیک به ۱ نشان‌دهنده عملکرد عالی.

AUC نزدیک به ۰.۵ نشان‌دهنده عملکردی مشابه پیش‌بینی تصادفی.

نتایج :

کلاس‌هایی که مدل در آن‌ها دقت بیشتری داشته مانند ragno یا farfalla دارای AUC بالاتری هستند.

کلاس‌هایی که مدل در آن‌ها ضعف دارد مانند gatto یا pecora دارای AUC پایین‌تری هستند.



فاز دوم:

پایه سازی این فاز دقیقاً همانند فاز قبل است تنها تفاوت در لایه ها و مدل است به جهت کاهش تعداد پارامتر ها به کمتر از ۴۰۰۰۰۰

کاهش تعداد فیلترهای لایه‌های کانولوشنی به صورت زیر است:

لایه اول: کاهش تعداد فیلترها از ۳۲ به ۸.

لایه دوم: کاهش تعداد فیلترها از ۶۴ به ۱۶.

لایه سوم: اضافه‌شده با ۳۲ فیلتر.

تحلیل معیارهای ارزیابی:

پرسیژن، ریکال و F1:

برای کلاس‌هایی مثل "ragno" و "farfalla"، همچنان عملکرد بالاست F1 نزدیک به ۰/۷ یا بالاتر.

عملکرد روی کلاس‌های دشوارتر مانند "gatto" و "elefante" همچنان ضعیف‌تر است، اما در برخی موارد مثل "mucca" و "pecora" نسبت به فاز اول بهبود مشاهده می‌شود.

میانگین وزنی: میانگین دقت، پرسیژن و ریکال حدود ۰/۶۰ است که اندکی بهتر از فاز اول است.

نتیجه گیری و مقایسه با فاز قبل:

با کاهش تعداد پارامترها، مدل در یادگیری و دقت تست به نتایج قابل قبولی دست یافته و عملکرد نسبت به فاز اول اندکی بهتر شده است. این نشان می‌دهد که طراحی بهینه‌تر می‌تواند حتی با تعداد پارامترهای کمتر به نتایج بهتری منجر شود.

ویژگی	فاز اول	فاز دوم
دقت یادگیری نهایی	98.71%	84.41%
دقت تست	57.44%	60.32%
تعداد پارامترها	بیشتر از 400,000	کمتر از 400,000
میانگین F1	0.58	0.60
عملکرد کلی روی تست	کمی پایین‌تر از حد انتظار	بهبود یافته

فاز سوم:

۱. Data Augmentation:

در این فاز برای جلوگیری از اورفیت (Overfitting)، از data augmentation استفاده کرده‌ام که این روش می‌تواند مدل را قادر به یادگیری ویژگی‌های عمومی‌تری از داده‌ها کند و به کاهش اورفیت کمک کند. روش‌های استفاده‌شده در data augmentation عبارتند از:

Random Horizontal Flip: که به‌طور تصادفی تصویر را به‌صورت افقی برعکس می‌کند.

Random Rotation (30 degrees): که به‌طور تصادفی تصویر را تا ۳۰ درجه می‌چرخاند.

Random Affine Transform: که تغییرات تصادفی در جابجایی (translates) تصاویر اعمال می‌کند.

این تغییرات باعث می‌شوند مدل تصاویر متنوع‌تری ببیند، که در نهایت به بهبود تعمیم‌پذیری (generalization) مدل کمک می‌کند.

۲. مدل:

در این فاز از مدل متفاوتی به نسبت فاز یک و دو استفاده کردم:

ویژگی‌های جدید مدل:

Batch Normalization: این لایه‌ها برای هر کانال ورودی میانگین و انحراف معیار را نرمال می‌کنند. این کار باعث می‌شود که مدل سریع‌تر یاد بگیرد و از vanishing gradient جلوگیری کند.

Dropout: از Dropout برای جلوگیری از اورفیت استفاده شده است. این لایه به‌طور تصادفی تعدادی از نورون‌ها را غیرفعال می‌کند تا مدل وابستگی بیش از حد به ویژگی‌های خاص را نداشته باشد.

سه لایه کانولوشن: شما از سه لایه کانولوشن با اندازه‌های مختلف فیلترها استفاده کرده‌اید تا ویژگی‌های پیچیده‌تری از تصاویر استخراج شود.

Fully connected layers: در انتهای شبکه، لایه‌های fully connected برای طبقه‌بندی نهایی اضافه شده است. از ReLU و Dropout برای جلوگیری از اورفیت استفاده شده است.

همچنین در این مرحله، از `CrossEntropyLoss` به عنوان تابع هزینه و از بهینه‌ساز `Adam` برای به‌روزرسانی وزن‌های مدل استفاده کردم چون این تابع هزینه برای مسائل طبقه‌بندی چندکلاسه مناسب است. این ترکیب برای آموزش مدل‌های طبقه‌بندی بسیار رایج است. بهینه‌ساز `Adam` کمک می‌کند که فرایند آموزش سریع‌تر و پایدارتر شود.

در این بخش، هدفم این بود که با استفاده از این تنظیمات آموزش مدل را به گونه‌ای انجام دهم که مدل بتواند به‌طور مؤثری ویژگی‌ها را یاد بگیرد و از اورفیتینگ جلوگیری شود.

۳. یادگیری و نرمال سازی:

در این بخش مدل را برای ۱۰ اپوک آموزش دادم. در هر اپوک، پس از انجام پیش‌بینی‌ها و محاسبه خطا با استفاده از تابع `CrossEntropyLoss`، عملیات به‌روزرسانی وزن‌ها را با `optimizer.step()` انجام دادم. در طول این فرایند، من به دقت مدل و ضرر (loss) آن پرداختم.

در ابتدا، دقت مدل حدود ۳۹/۱۵٪ بود و با گذشت زمان و کاهش مقدار خطا، دقت به ۶۳/۵۴٪ در پایان اپوک دهم رسید.

ضرر مدل نیز در طول زمان کاهش پیدا کرد، که نشان‌دهنده بهبود عملکرد مدل است.

همچنین دقت تست به ۶۲/۸۰٪ رسید اختلاف کم دقت بخش یادگیری و تست نشان دهنده عدم اورفیت مجموعه است.

۴. معیار های ارزیابی:

نتایج ارزیابی مدل در فاز سه نشان می‌دهد که دقت کلی مدل بر روی داده‌های تست ۶۳/۰۰٪ است. در مقایسه با فازهای قبلی، این دقت نسبت به فاز ۲ که ۶۰/۳۲٪ بود، کمی بهبود یافته است.

در گزارش طبقه‌بندی (Classification Report) شاهد برخی تفاوت‌ها در معیارهای دقت (precision)، (recall) و F1-score هستیم:

Precision و recall به طور کلی بهبود یافته‌اند، به ویژه برای دسته‌هایی مثل "cane" و "ragno".

F1-score که میانگین دقت و یادآوری است، برای دسته‌هایی مانند "farfalla" و "gallina" افزایش یافته است، که نشان می‌دهد مدل در این دسته‌ها بهتر عمل کرده است.

به طور کلی، نتایج میانگین وزن‌دار و ماکرو هم بهبود را نشان می‌دهند.

این تغییرات به احتمال زیاد ناشی از استفاده از تکنیک‌هایی مانند `data augmentation`، `batch` و `dropout`، `normalization` است که توانسته‌اند از `overfitting` جلوگیری کنند و مدل را بهبود بخشند. با این حال، برخی دسته‌ها مانند "pecora" و "gatto" هنوز دقت و یادآوری نسبتاً پایین دارند.

ادامه بخش های فاز سه همانند فاز های قبلی است.