**گزارش مدل کد ارسال شده**

**1. مقدمه**

در این پروژه، یک مدل شبکه عصبی برای شناسایی تصاویر سگ و گربه با استفاده از روش‌های پیشرفته یادگیری عمیق (Deep Learning) پیاده‌سازی شده است. برای این منظور از چندین مدل پیش‌آماده (Pre-trained) برای استخراج ویژگی‌ها استفاده شده است. به دلیل اینکه برای شناسایی تصاویر از مدل MLP استفاده نکردیم، بلکه از مدل‌های پیشرفته‌تر به عنوان شتاب‌دهنده استفاده شد، در این گزارش روش‌ها و مراحل انجام کار را توضیح خواهیم داد.

**2. کتابخانه‌ها و پیکربندی‌های اولیه**

ابتدا برای انجام پردازش‌ها و آموزش مدل از چندین کتابخانه در پایتون استفاده شد:

* **Numpy** و **Matplotlib** برای پردازش داده‌ها و نمایش گراف‌ها.
* **TensorFlow** و **Keras** برای طراحی و آموزش مدل‌ها.
* **Pandas** و **Seaborn** برای تحلیل داده‌ها و رسم نمودارها.

**3. پیش‌پردازش داده‌ها و آماده‌سازی آن‌ها**

در ابتدا داده‌های تصاویر از دایرکتوری‌های مشخص‌شده (train\_set و test\_set) بارگیری شدند. سپس از **ImageDataGenerator** برای اعمال تقویت داده‌ها (Data Augmentation) به مجموعه آموزش استفاده شد. این عملیات شامل چرخش، تغییر اندازه، برش عرضی، تغییر ارتفاع، و برگشت افقی تصاویر است.

به‌منظور بهبود عملکرد مدل، تصاویر به مقیاس [0, 1] تغییر داده شدند و از شبیه‌سازی تغییرات موجود در داده‌ها استفاده شد.

train\_datagen = ImageDataGenerator(

rescale=1.0/255,

rotation\_range=20,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

horizontal\_flip=True,

fill\_mode='nearest'

)

**4. ساخت مدل‌های انتقال یادگیری (Transfer Learning)**

در این بخش از مدل‌های از پیش‌آماده (pre-trained) مانند ResNet50V2، DenseNet121 و Xception برای استخراج ویژگی‌ها از تصاویر استفاده شد. این مدل‌ها وزن‌های خود را از مجموعه داده **ImageNet** به ارث برده‌اند. این مدل‌ها در حالت **غیر قابل آموزش** (frozen) قرار گرفتند تا فقط به‌عنوان استخراج‌کننده ویژگی‌ها از آن‌ها استفاده شود.

TL\_Models = [

ResNet50V2(input\_shape=(images\_size, images\_size, 3), weights='imagenet', include\_top=False),

DenseNet121(input\_shape=(images\_size, images\_size, 3), weights='imagenet', include\_top=False),

Xception(input\_shape=(images\_size, images\_size, 3), weights='imagenet', include\_top=False)

]

**5. آموزش مدل‌ها**

در این مرحله، برای هر کدام از مدل‌های انتقال یادگیری، یک معماری ساده شبکه عصبی با یک لایه **GlobalAveragePooling2D** برای فشرده‌سازی ویژگی‌ها و یک لایه **Dropout** برای جلوگیری از اورفیتینگ اضافه شد. سپس از یک لایه **Dense** برای طبقه‌بندی تصاویر به دو کلاس (سگ و گربه) استفاده شد.

این مدل‌ها با استفاده از **Adam Optimizer** و **Categorical Crossentropy Loss** آموزش داده شدند.

model = keras.Sequential([

tl\_model,

layers.GlobalAveragePooling2D(),

layers.Dropout(0.5),

layers.Dense(2, activation='softmax')

])

**6. تعیین کاهش نرخ یادگیری (Learning Rate)**

برای بهینه‌سازی عملکرد مدل، از دو روش برای کاهش نرخ یادگیری استفاده شد:

1. **کاهش نرخ یادگیری به‌صورت خودکار** زمانی که عملکرد مدل بهبود نمی‌یابد (با استفاده از ReduceLROnPlateau).
2. **برنامه زمان‌بندی نرخ یادگیری ExponentialDecay** که نرخ یادگیری را به‌صورت نمایی کاهش می‌دهد.

learning\_rate\_schedule = keras.optimizers.schedules.ExponentialDecay(

initial\_learning\_rate=0.01,

decay\_steps=1000,

decay\_rate=0.5,

min\_lr=0.00001

)

**7. آموزش نهایی مدل**

در مرحله نهایی، پس از انتخاب بهترین مدل، مدل نهایی برای 20 اپوک به کمک داده‌های آموزش و اعتبارسنجی آموزش داده شد. همچنین از تکنیک **EarlyStopping** برای جلوگیری از اورفیتینگ و **LearningRateScheduler** برای کاهش نرخ یادگیری استفاده شد.

history = model.fit(

train\_generator,

steps\_per\_epoch=train\_generator.samples // batch\_size,

epochs=20,

validation\_data=test\_generator,

validation\_steps=test\_generator.samples // batch\_size,

callbacks=[callback]

)

**8. ارزیابی مدل**

پس از آموزش، مدل روی داده‌های آزمون ارزیابی شد و دقت و خطای آن در نمودارهای مختلف نشان داده شد. همچنین ماتریس سردرگمی (Confusion Matrix) برای تحلیل بهتر پیش‌بینی‌های مدل ترسیم شد.

**9. نتیجه‌گیری**

در این پروژه، از مدل‌های پیشرفته یادگیری عمیق برای شناسایی تصاویر سگ و گربه استفاده شد. به دلیل ویژگی‌های پیچیده تصاویر، تنها از مدل‌های **MLP** استفاده نشد زیرا برای داده‌های تصویری مناسب نیستند. استفاده از **مدل‌های انتقال یادگیری** مانند ResNet و DenseNet منجر به نتایج بهتری شد.

تمامی مراحل شامل پیش‌پردازش داده‌ها، ایجاد و آموزش مدل‌ها، تنظیم نرخ یادگیری و ارزیابی نتایج به‌طور دقیق پیاده‌سازی شد.

این رویکرد باعث بهبود دقت مدل و کاهش خطاها در شناسایی تصاویر می‌شود.