

پروژه نهایی هوش محاسباتی

1- امیر عباسی (960122680040)

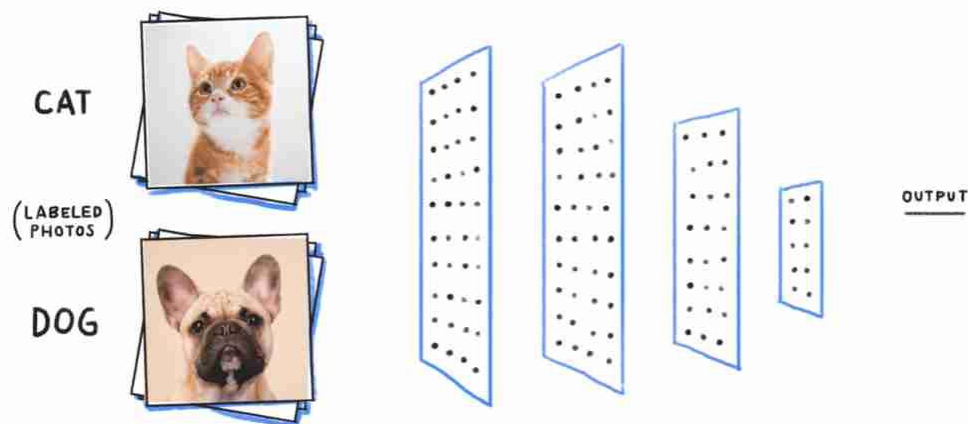
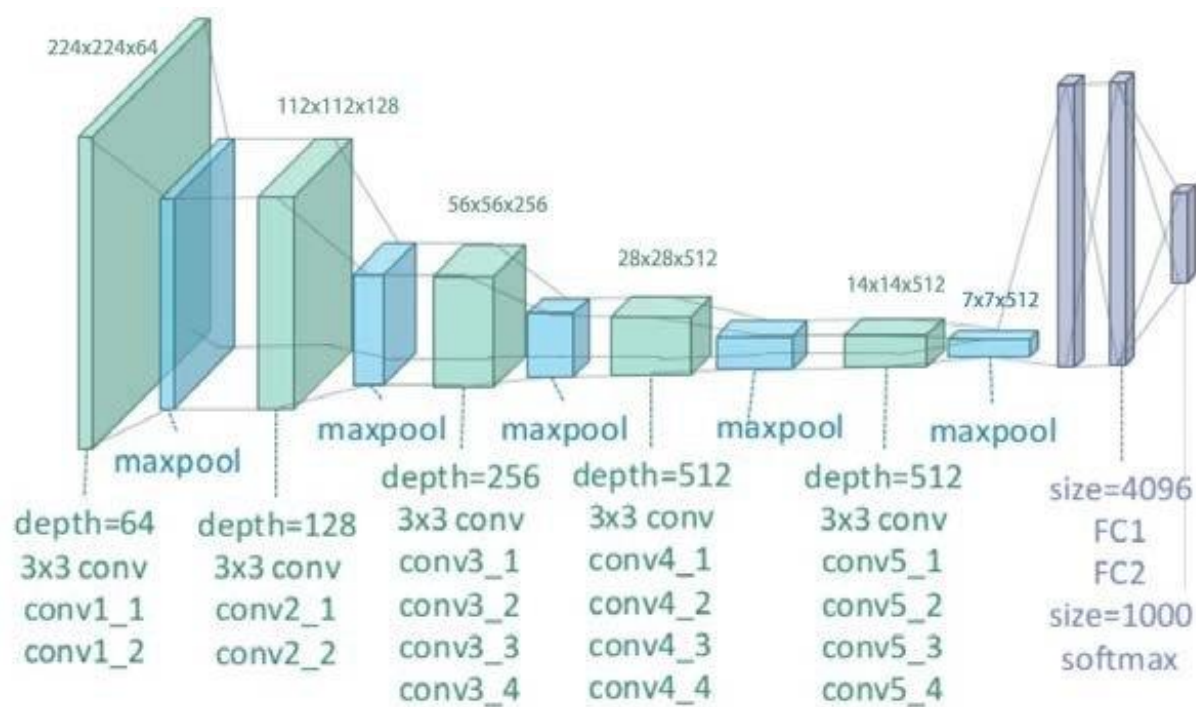
2- رضا خان محمدی (96122680001)

3- حسین اعتدادی (960122680004)

4- آرین بهمنی (960122680085)

Transfer Learning using Convolutional Neural Networks for Classifying Dogs and Cats

در این پروژه قصد داریم به کمک شبکه عصبی پیچشی یا همان Convolutional Neural Network مدلی طراحی کنیم تا بتواند عکس های سگ و گربه را از همدیگر تشخیص دهد. برای طراحی این پروژه از سیستم انتقال یادگیری یا همان Transfer Learning به کمک مدل از پیش تربیت شده vgg19، استفاده می کنیم، تمامی پیاده سازی ها به کمک کتابخانه Tensorflow و Keras، دو تا از مهمترین و قدرتمند ترین کتابخانه های شبکه های عصبی و به خصوص برای پروژه های بینایی ماشین انجام شده.



دیتاست مورد استفاده شده در این پروژه برگرفته شده از مایکروسافت، شامل 24000 عکس سگ و گربه است که در مرحله آموزش این مدل 1000 عکس برای گربه، و به همین مقدار برای سگ در نظر گرفته شده که هرکدام 224×224 بوده و در سه لایه RGB در کنار هم قرار گرفته اند.، از این دیتاست داده های مرحله Cross-Validation و Test هم جدا می شود.

```
[ ] !wget "https://download.microsoft.com/download/3/E/1/3E1C3F21-ECDB-4869-8368-6DEBA77B919F/kagglecatsanddogs_3367a.zip"
!unzip "/content/kagglecatsanddogs_3367a.zip"
!rm "/content/PetImages/Cat/Thumbs.db"
!rm "/content/PetImages/Dog/Thumbs.db"
address_cats="/content/PetImages/Cat"
address_dogs="/content/PetImages/Dog"
```

حال به طراحی مدل و آموزش آن می پردازیم، برای اینکار ابتدا لایه آخر vgg19 (لایه خروجی) را حذف میکنیم و محتویات آن را Flatten میکنیم و یک لایه Dense با 1024 نورون با تابع فعالسازی Selu به انتهای آن اضافه میکنیم و پس از آن لایه خروجی را هم یک لایه Dense تک نورونه با تابع فعالسازی Sigmoid برای گرفتن خروجی به صورت احتمال در نظر می گیریم، از آنجایی که عمل اصلی که در آموزش مدل استفاده میشود مقداردهی وزن ها در هر مرحله است بایستی تنها به آموزش لایه آخری که به vgg19 اضافه میکنیم پردازیم و دیگر لایه ها و وزن هایشان را Freeze کنیم. بهینه ساز این مدل Adam بوده که در 50 epoch مراحل آموزش را میگذراند.

```
[ ] model=keras.applications.vgg19.VGG19(include_top=False, weights='imagenet', input_shape=(224,224,3))
for i in model.layers:
    i.trainable=False
x = Flatten()(model.layers[-1].output)
x = Dense(1024, activation="selu")(x)
x=Dense(1, activation="sigmoid")(x)
model=keras.Model(model.input,x)

[ ] checkpoint = ModelCheckpoint("w.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
model.compile(optimizer="adam", loss="binary_crossentropy", metrics=['accuracy'])
model.fit(x=x_train,y=y_train,epochs=50,validation_data=(x_val,y_val),batch_size=256,callbacks=[checkpoint])
```

و بدین صورت مدل ما آموزش داده می شود که با فراخوانی دستور `fit`. بر روی مدل می توانیم تعداد 1992 تا داده Test را بیازماییم. Result نهایی بدست آمده در این پروژه 89% می باشد.

```
[ ] model.evaluate(x_test,y_test)
```

```
1992/1992 [=====] - 7s 3ms/sample - loss: 0.2689 - acc: 0.8916  
[0.2688913242704418, 0.8915663]
```

