

بخش 1:

(الف)

2D-convolution: داده های ورودی 3 بعد دارد طول و عرض و تعداد کانال ها. (feature map مثل: rgb) هر کانال بر جنبه های مختلف تصویر اصلی تاکید دارد به اینصورت که یک کرنل تعریف می شود که وزن داده های دوبعدی (طول و عرض) را برای هر کانال تعریف می کند این convolution اغلب برای استخراج ویژگی از تصاویر استفاده می شود مثل:

image classification ,face recognition

3D-convolution: داده های ورودی 4 بعد دارد طول و عرض تعداد کانال ها و عمق .

در واقع 3d convolution در واقع تعمیم یافته 2d convolution هست فیلتر 3d در هر 3 جهت (طول و عرض و کانال) حرکت میکند که در هر موقعیت که با ضرب و جمع (element-wise multiplication and addition) یک عدد تولید می کند از کاربر های این مدل میتوان به 3d medical images, event detection in video

منبع: [towardsdatascience](https://towardsdatascience.com/3d-convolution-for-video-analysis-7a1e1e1e1e1e)

(ب)

به طور معمول از ماتریس مربعی استفاده می شود چرا که هم میتوان آن را در شرایط خاص مثل تشخیص لبه ای افقی و عمودی آن را به ماتریس مستطیلی تعریف کرد و همچنین یک سری عملیات ریاضی بر روی ماتریس های مربعی قابل انجام است و اگر ماتریس مربعی متقارن باشد نیز بهتر است تا محدودیت خاصی در یک جهت خاص ایجاد نکند.

برای تعیین سایز کرنل/فیلتر بهتر است زیاد نباشد چون یکی از دلایل استفاده از cnn ها اشتراک وزن ها و کاهش محاسبات هست و از این مهمتر اینکه به دنبال استخراج ویژگی های محلی هستیم که اگر سایز کرنل با توجه به سایز ورودی نسبتا بزرگ باشد امکان استخراج این ویژگی ها را ندارد.

در حالت کلی کرنل با سایز کوچک تر ویژگی های جزیی تری استخراج میکند و با سایز بزرگ تر ویژگی های کلی تر که بهتر است در لایه های اولیه کانولوشنی از کرنل با سایز بزرگ تر استفاده کنیم و تا ابتدا ویژگی های کلی تر را استخراج کنیم سپس سایز کرنل را کمتر کنیم سایز کرنل بسیار وابسته به سایز تصویر ورودی است .

(ج)

در max pooling بزرگ ترین مقداری که در کرنل وجود دارد را انتخاب می کنیم به این منظور که ان مقدار تاثیر بیشتری دارد و باقی کانال ها تاثیر ناچیزی یا مخربی برای استخراج ویژگی دارند

در min pooling برعکس max pooling کوچک ترین مقدار را انتخاب می کنیم از لحاظ کاربر شبیه هم هست که بسیار وابسته به داده های ما هست مثلا اگر مقدار پیکسل ها را ریورس کنیم min pooling همان تاثیر max pooling را دارد در average pooling می ایم میانگین هر کرنل را حساب می کنیم تا تاثیر همه موقعیت ها حفظ شود همچنین کاهش ابعاد را نیز داشته باشیم در adaptive pooling ما فقط ابعاد خروجی را می دهیم و hyper parameter ها در خود مدل آموزش داده می شود

بخش 2:

عکس اصلی در gray scale



فیلتر blur که باعث می شود نویز عکس کم شود با جمع کردن با مقدار یکسان پیکسل ها به انداز کرنل سایز

```
box_blur = (1/9) * np.array([
    [1, 1, 1],
    [1, 1, 1],
    [1, 1, 1]])
```



فیلتر edge detector تمام لبه ها را تشخیص میدهد

```
edge_detector = np.array([
    [-1, -1, -1],
    [-1, 8, -1],
    [-1, -1, -1]])
```



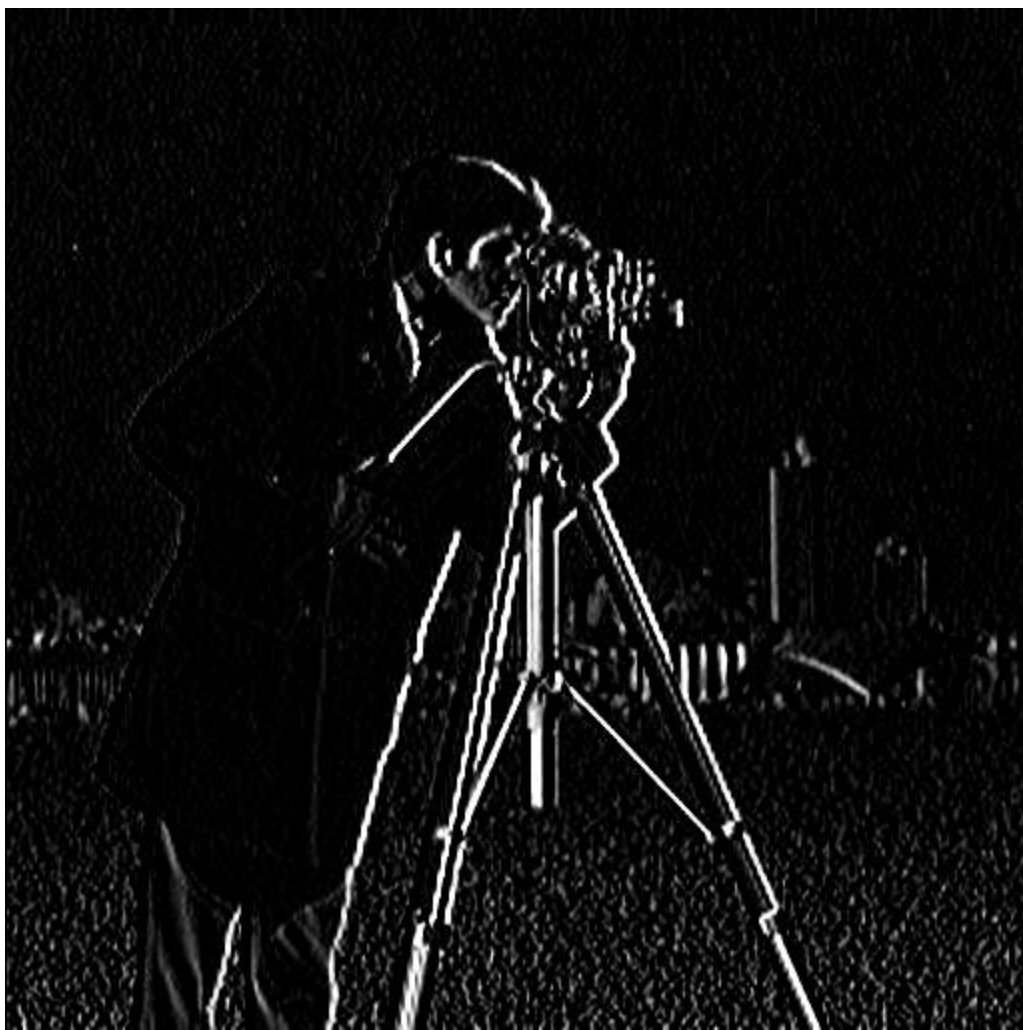
فیلتر `sobel_y` می شود اختلاف پیکسل های پایین از بالا برای یافتن لبه های افقی

```
sobel_y = np.array(  
    [[-1, -2, -1],  
     [0, 0, 0],  
     [1, 2, 1]])
```



فیلتر `sobel_x` می شود اختلاف پیکسل های راست از چپ برای یافتن لبه های عمودی

```
sobel_x = np.array(  
    [[-1, 0, 1],  
     [-2, 0, 2],  
     [-1, 0, 1]])
```



بخش 3)

الف) keras tuner به ما کمک میکند تا بهترین hyper parameter ها را پیدا کنیم در حالت عادی ما بصورت دستی و a/b testing به انتخاب hyper parameter ها می پرداختیم که اتلاف زمان زیادی داشت و همینطور ممکن بود بهترین ان ها را نیز پیدا نکنیم اما با keras tuner می توانیم hyper parameter ها را tune کنیم

ب)

RandomSearch Tuner: در این مدل یک فضای جستجو وجود دارد که به صورت رندوم از این فضا hyper parameter ها را بهینه می کند

Hyperband Tuner: این مدل بهبود یافته randomsearch tuner است با این تفاوت در random search بعضی از hyper parameter ها در چند epoch اول مشهود است که بهینه نمی شود اما تا آخر ادامه میدهد اما در hyperband از early stopping استفاده می کند تا مدل با این hyperparameter بد tune نشود .

BayesianOptimization Tuner: مدلی است که به صورت احتمالی hyper parameter ها را به score احتمالی objective function وصل می کند و برخلاف دو مدل قبل که از hyperparameter هایی که در آزمایش های قبل تنظیم شده استفاده نکردند از آنها برای ایجاد مدل احتمالی جدید استفاده می کند. در این سوال از این مدل استفاده می کنیم چون از hyperparameters های مدل هایی که قبلاً evaluate شده استفاده می کند.

منبع: [towardsdatascience](https://towardsdatascience.com/)

(ج)

```
[35] def build_model(hp):
    model = keras.Sequential()
    # Tune the number of layers.
    for i in range(hp.Int("cn_num_layers", 2, 5)):
        model.add(
            layers.Conv2D(
                # Tune number of units separately.
                filters=hp.Int(f"filters_{i}", min_value=32, max_value=256, step=16),
                kernel_size=(3, 3),
                activation=hp.Choice("cn_activation", ["relu", "tanh"]),
            )
        )
    if hp.Boolean("cn_dropout"):
        model.add(layers.Dropout(rate=0.25))

    model.add(layers.Flatten())
    for i in range(hp.Int("num_layers", 1, 5)):
        model.add(
            layers.Dense(
                # Tune number of units separately.
                units=hp.Int(f"units_{i}", min_value=32, max_value=256, step=32),
                activation=hp.Choice("activation", ["relu", "tanh"]),
            )
        )
    if hp.Boolean("dropout"):
        model.add(layers.Dropout(rate=hp.Float("dropout_rate", min_value=0, max_value=0.7, step=0.1)))
    model.add(layers.Dense(10, activation="softmax"))

    learning_rate = hp.Float("lr", min_value=1e-4, max_value=1e-2, sampling="log")
    model.compile(
        optimizer=keras.optimizers.Adam(learning_rate=learning_rate),
        loss="categorical_crossentropy",
        metrics=["accuracy"],
    )
    return model
```

با مشخصات بالا که غالب hyperparameter را به عهده keras tuner دادیم و BayesianOptimization مقدار hyperparameters ها را بهینه کردیم

```
[36] tuner = kt.BayesianOptimization(
    build_model,
    objective="val_accuracy",
    max_trials=5,
    num_initial_points=2,
    seed=np.random.seed(42)
)
```

در نهایت در بهترین آزمایش به دقت 70 درصد رسیدیم

```
tuner.search(
    x_train, y_train,
    epochs=15,
    validation_data=(x_test, y_test)
)
```

```
Trial 5 Complete [00h 02m 24s]
val_accuracy: 0.6690000295639038
```

```
Best val_accuracy So Far: 0.7059999704360962
Total elapsed time: 00h 11m 36s
INFO:tensorflow:Oracle triggered exit
```