پاسخ سوال اول

قسمت الف:

هر پیکسل در آخرین چهارم با 9 پیکسل در لایه سوم در ارتباط است. به سادگی میتوان دید با طول گام ثابت 1 برای یک پنجره به طول n*n در پنجره قبل اگر اندازه پنجره k*k باشد به میزان a*a که a=k+n-1 پیکسل مورد مشاهده قرار میگیرند.

لایه چهارم به سوم n=1,k=3 بنابراین :a = 3

لایه سوم به دوم : n=3,k=3 بنابراین: a = 5

لایه دوم به اول : n=5, k=3 بنابراین: a = 7

بنابراین هر پیکسل در لایه آخر به 49 پیکسل لایه اول مربوط است.

اجتماع آن ها یعنی 9+25+49 =83

قسمت ب:

این لایه بعد را در کاهش میدهد و به ما کمک میکند scale های مختلف تصویر را با فیلتر های مختلف مشاهده کنیم. در این کار سعی میشود اطلاعات ارزشمندتر استخراج شده حفظ و باقی آن ها حذف شود. معمولا پس از این کار تعداد پارامتر ها را با افزایش تعداد فیلتر ها افزایش میدهند و در نهایت یک شبکه عمیقتر با تعداد پارامتر مناسب خواهیم داشت. مشکل دیگری که حل میکند این است که تصاویر ما ممکن است مقداری جابجایی مکانی داشته باشد. هنگامی که این لایه اضافه میشود برای بازه ای مشکل این مقدار جابجایی حل شده است چون از همه آن patch صرفا یک عدد اعلام شده است.

قسمت ج:

چون same است پس :$32 * 35 * 35$ ابعاد خروجی است.

تعداد پارامتر ها برابر است با: $32 * (3 * 3 * 16 + 1) = 4640$

قسمت د:

یعنی $(35 * 35 * 32) * (35 * 35 * 16 + 1)$

اگر تعداد فیلتر ها را درنظر نگیریم و فقط سایز تصویر مهم باشد: $(35 * 35) * (35 * 35 + 1)$

بدون محاسبه نیز به سادگی دیده میشود که حاصل بالا خیلی خیلی بیشتر از حالت قبل است.

قسمت هـ :

شبکه را در محیط کراس پیاده کردم و اندازه خروجی و تعداد پارامتر ها به شرح زیر است:

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 223, 223, 128)     9728
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 219, 219, 34)      108834
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 108, 108, 34)      0
_________________________________________________________________
dense (Dense)                (None, 108, 108, 512)     17920
```

```
_________________________________________________________________
dense_1 (Dense)              (None, 108, 108, 100)     51300
=================================================================
Total params: 187,782
Trainable params: 187,782
Non-trainable params: 0
_________________________________________________________________
```

این شبکه با عمق کم تعداد پارامتر های زیادی دارد. برای حل این مشکل میتوان فکتوریزیشن های زیر را انجام داد

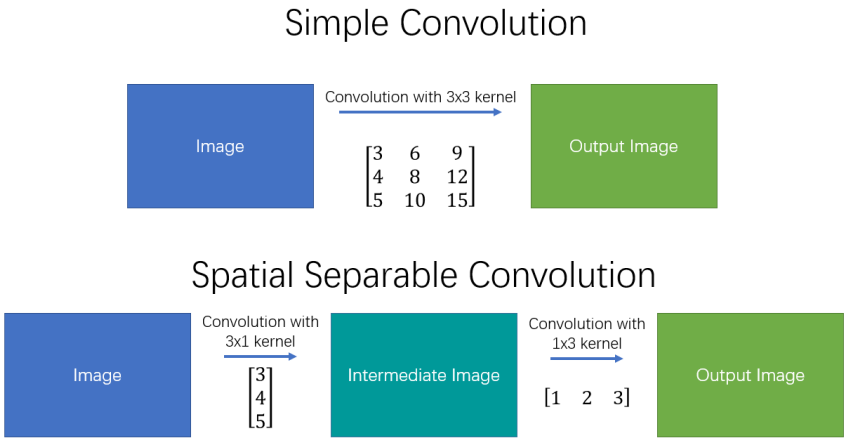
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 225, 225, 64)      1792
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 223, 223, 64)      36928
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 221, 221, 17)      9809
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 219, 219, 17)      2618
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 108, 108, 17)      0
_________________________________________________________________
dense_4 (Dense)              (None, 108, 108, 512)     9216
_________________________________________________________________
dense_5 (Dense)              (None, 108, 108, 100)     51300
=================================================================
Total params: 111,663
Trainable params: 111,663
Non-trainable params: 0
_________________________________________________________________
```

حالت دیگر

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_10 (Conv2D)           (None, 225, 227, 32)      320
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 225, 225, 32)      3104
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 223, 225, 32)      3104
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 223, 223, 32)      3104
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 221, 221, 17)      4913
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 219, 219, 17)      2618
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 108, 108, 17)      0
_________________________________________________________________
dense_6 (Dense)              (None, 108, 108, 512)     9216
_________________________________________________________________
```

```
dense_7 (Dense)              (None, 108, 108, 100)     51300
=================================================================
Total params: 77,679
Trainable params: 77,679
Non-trainable params: 0
_________________________________________________________________
```

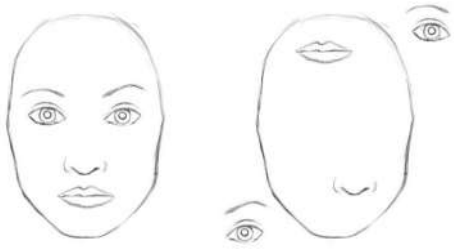
قسمت ز:



در این ساختار با استفاده بجای استفاده از 9 پارامتر از 6 پارامتر به شکل بالا استفاده میکنند. گرچه درجه آزادی و تعداد پارامتر ها کاهش می یابد ولی عمق افزایش می یابد و خاصیت غیر خطی شبکه را بیشتر میکند.

قسمت ح:

در شبکه کانولوشنی با استفاده از لایه ها ویژگی های لول پایین را استخراج میکنند و در لایه های بالا تر با ترکیب این ها ویژگی های پیچیده تری را مانند شناسایی آبجکت های مختلف میکنند و وقتی این آبجکت ها به لایه تمام متصل میرسد بایکدیگر آمیخته میشوند و به پاسخ می رسند. اما در این روش ها خود آبجکت بیشتر اهمیت دارد تا مکان و زوایه آنها نسبت به یکدیگر. مثلا شبکه کانولوشنی ممکن است هردو شکل را چهره تشخیص دهد زیرا صرف داشتن اجزای صورت برای آن کفایت میکند.



برای حل این مشکل جفری هینتون ابتدا به بیان ایجاد مشکل میپردازد و سپس شبکه کپسولی خود را مطرح میکند که ایده اصلی آن از رندر کردن آبجکت های مختلف در کارهای گرافیکی کامپیوتر اما به شکل معکوس آن است. مورد دیگر مثلا تصویر زیر را در نظر بگیرد. اگر انسان تنها یک بار تصویر مجسمه آزادی را دیده باشد میتواند تشخیص دهد که تمامی تصاویر زیر یکی است ولی از زوایای مختلف. در حقیقت این جابجایی و چرخش در تشخیص ما مشکل ساز نمی شوند. زیرا ما رابطه مختلف اشیا را در نظر گرفته ایم و آن را ذخیره میکنیم.

یادگیری این تصویر برای انسان با کمتر از انگشت های دست دیدن آن ممکن است. برای شبکه کپسولی شاید با چند صد تصویر ولی برای شبکه کانولوشنی ده ها هزار تصویر با جابجایی ها و زوایای مختلف نیاز است تا آن را شناسایی کند.