机器学习纳米学位

毕业项目

吴品曄 优达学城

2018年5月13日

I. 问题的定义

项目概述

我決定加入機器學習工程師Nanodegree的原因之一是為了讓生活更美好。我選擇了 distracted driver detection主題作為我的畢業項目，因為我想幫助避免因分心駕駛造成的悲劇。我們都曾經站在那個路口：當紅綠燈變綠了，你面前的車沒有退讓。或者，一輛車突然出現並開始左右擺動。當你望向眼前的司機時，你發現司機正在發短信，與乘客交談，或者拿著手機與話筒的另一端有說有笑，而你並沒有感到驚訝。

當車輛的時速為每小時90公里時，發送或閱讀短信會讓您的視線離開道路約5秒，足以行駛過一座美式足球場。根據CDC motor vehicle safety divisio[1]的報導，在2015年，有百分之十六的交通事故是走神駕駛造成的。State Farm與Kaggle合作舉辦了一場比賽[2]要求挑戰者使用機器學習方法提出針對走神司機的感測模型以應用在他們的儀表板相機產品，希望透過更好的警報系統保障使用者與行人的人身安全。這個項目屬於計算機視覺領域，教機器識別人類的動作，此一領域相關的數據集有UCF101、DeepMind的Kinetics和Google的AVA等等。這份報告目標運用深度學習建立一個根據訓練圖片學習走神駕駛的特徵進而檢測出走神司機的模型。

问题陈述

這個問題屬於機器學習中的多類分類問題，需要讓機器識別圖像中的駕駛的狀態以改善對於分心行為的警報系統。我們將駕駛分為10種狀態，其中只有1種是安全駕駛，其他則是9種不同類型的分心駕駛。目標是從提供的分類中預測駕駛員處於某個分類中的可能性。

近年來卷積神經網絡（CNN）在計算機視覺領域取得了重大的突破，這個項目也將使用卷積神經網絡，訓練模型根據輸入的駕駛圖片判斷圖中駕駛員的狀態屬於每一種分類的機率。我們期望模型對於最接近正確答案的分類能給出一個接近1的預測值，同時對其他非正確答案的分類給出接近0的值，代表模型能找出最接近的答案並對預測結果有足夠的信心。

评价指标

Kaggle在本次比賽中使用的評估指標是 multi-class logarithmic loss。每張圖片都標有一個真實的分類。模型將針對每個圖像屬於每個分類的可能性提出一組預測，並通過以下公式評估性能，

$$logloss = \frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(p_{ij})$$

N表示測試集中的圖片總數，M表示對於圖片所有可能的分類數，log 使用的是natural logarithm，當圖片i 屬於類別 j 時，y_{ij} 等於1，反之則為0。p_{ij} 代表模型預測圖片i 屬於類別j 的機率。為了避免log函數的極端情況， Kaggle官方作法將預測機率由$max(min(p, 1-10^{-15}), 10^{-15})$ 取代。

Multi-class logarithmic loss 會向未對正確答案具有足夠信心的分類器進行懲罰。當y 的值為1而對應的p 也接近1， $log(p)$ 的值趨近於0，log loss將會是很小的值；如果對應的p 接近0，$log(p)$ 將會是很大的負值，根據公式將產生不小的log loss。使用logarithmic loss較accuracy的好處在於：在每種分類樣本比例分布不平衡的情況下，accuracy指標會發生accuracy paradox。例如在一個二分類問題，數據集中有99個正例與1個反例，一個只輸出正例的分類器可以得到99%的準確率，然而此時accuracy僅僅反映了數據潛在的類別分布，而logarithmic loss是一種maximum likelihood estimation，既使在樣本比例分布不平衡的情況下仍能反映模型對每個樣本分類概率的估計。

II. 分析

数据的探索

數據集是由State Farm提供的司機圖像，大小為640X480，每個圖像中的司機在車上進行某種操作（安全駕駛，發短信，打電話，喝飲料，伸手到後方等等）。訓練集與測試集的司機是分開的，一位司機只能出現在訓練集或測試集上。圖像的司機皆為左駕，由特定的視角拍攝，我們將需要讓模型從圖象中觀察手、司機的視線、身體位置，手機或飲料的位置等特徵判斷司機所屬的狀態。

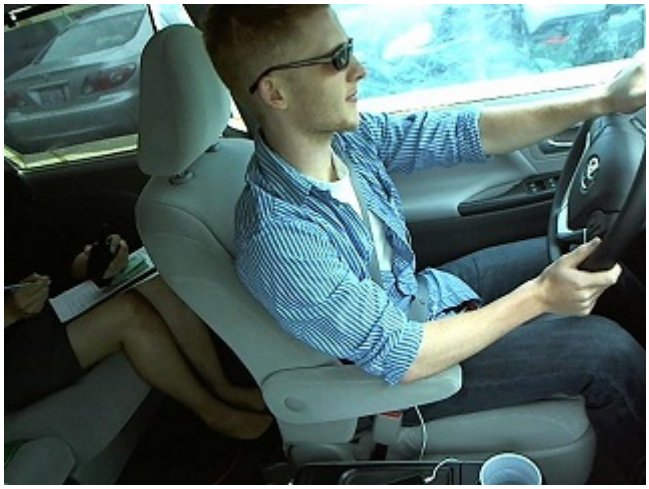

圖1： 典型的輸入樣例

下圖為數據集中每個分類樣本數的直方圖：

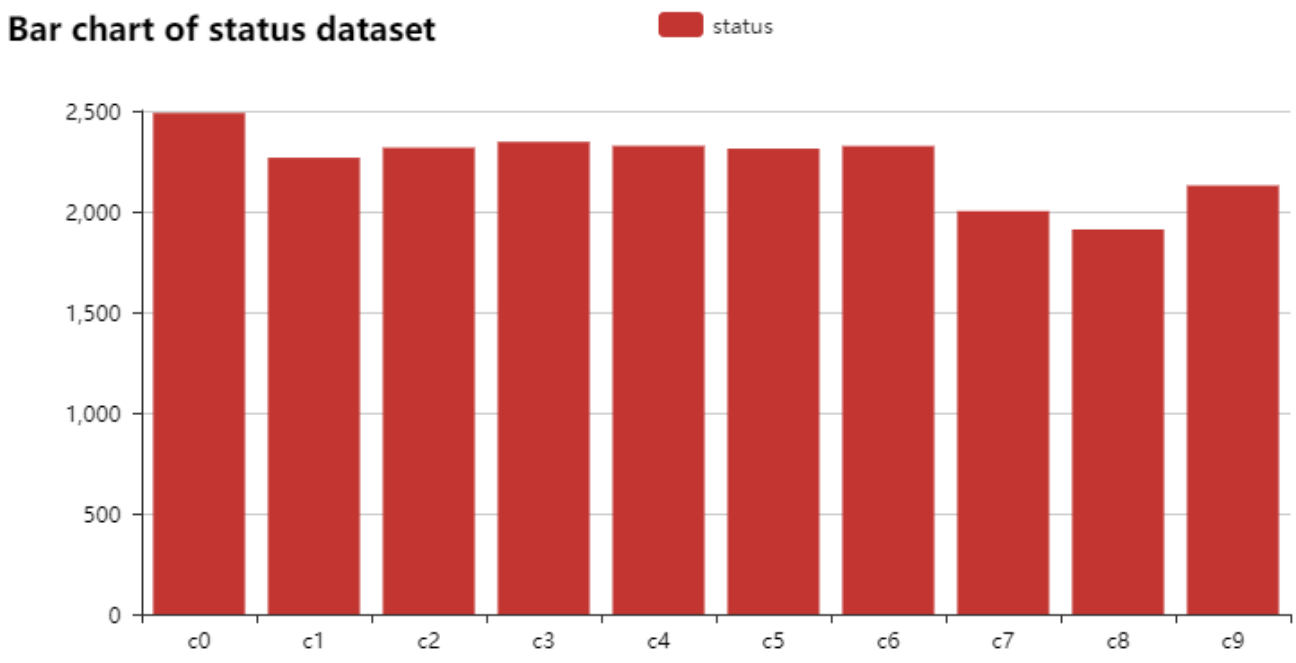

圖2： Bar chart of status dataset

樣本數最多的c0(安全駕駛)共有2,489筆資料，佔訓練集的11.10%；樣本數最少的c8(化妝，整理頭髮)共有1911筆資料，佔訓練集的8.52%。最多與最少的樣本差距為2.58%，標準差只有0.74%，表示訓練集具有不錯的平衡性。

為了模擬目標情境並確保駕駛安全，這些圖像在設置環境中拍攝的。由一輛卡車拖著汽車在街上行駛，因此這些「司機」並非真的在駕駛。

由於此項目的數據集取自若干個影片，許多鄰近的幀幾乎可視為一模一樣的圖像，若是連續幾張圖像分別放在訓練及與驗證集裡可能讓模型只記住圖片的長相而未學會分析司機的狀態，導致驗證集的分數高但是不可信，因此不能對數據集做shuffle。此項目有提供標記司機ID與圖像的CSV檔案，我們可以用司機ID來劃分訓練集與驗證集。

算法和技术

這個項目裡我使用過的模型有VGG16[3]、VGG19、ResNet50[4]、Inception V3[5]、Xception[6]、InceptionResNet V2[7]以及DenseNet[8]這些在圖像識別領域取得重大成果的經典模型，以下分別探討這幾個模型的技術以及訓練模型將常使用的Adam優化算法[9]：

- VGG
  VGG網路架構於2014年出現在Simonyan和Zisserman的論文中,在ILSVRC-2014分類挑戰中取得第二名的結果。圖像輸入尺寸為224x224，預處理將每個pixel減去訓練集中的RGB均值。使用3X3的filters，stride與padding皆為1個pixel，空間池化由5個2X2最大池化層進行，步長為2。圖3為VGG16與VGG19的架構。

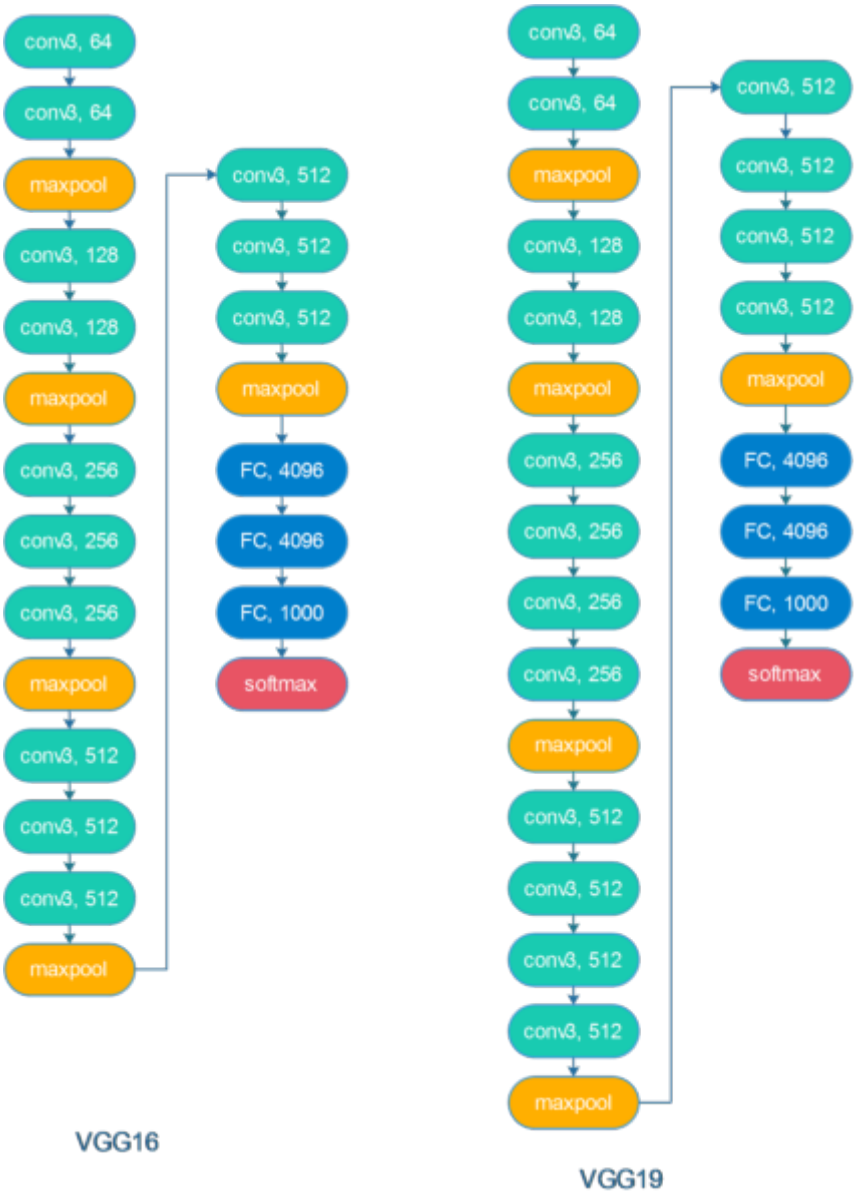

圖3： VGG16 & VGG19

- ResNet
  隨著CNN的深度加深，較複雜的模型往往更難以優化，訓練模型可能會出現training accuracy的degradation問題。ResNet引入深度殘差學習框架作為解決方案，將期望的基本映射表示為$H(x)$，非線性層擬合另一個映射$F(x) := H(x) - x$，原本的映射重寫為$F(x) + x$。在極端情況下，如果恆等映射為最優，將殘差設置為0會比通過一堆非線性層擬合恆等映射更容易。圖4為帶有快捷連接的神經網路結構，構建塊個公式定義為$y = F(x, W_i) + x$，x和y分別代表輸入和輸出，$F(x, W_i)$則是要學習的殘差映射，$F = W_2\sigma(W_1x)$，σ表示ReLU。x和F的維度必須是相等的，若不相等則透過快捷連結進行線性投射W_S匹配維度:$y = F(x, W_i) + W_sx$。ResNet可以建構超過1000層的模型，在2015年的ILSVRC與COCO競賽皆取得第1名的成績。

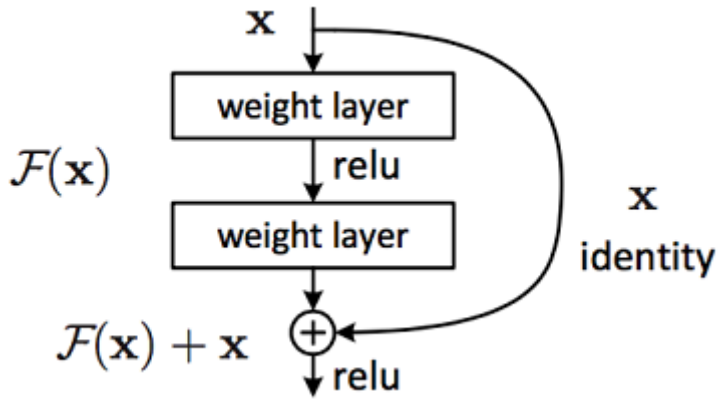

圖4： a residual learning building block

- Inception V3 & InceptionResNet V2
  Inception是GoogLeNet提出的CNN結構，主要想法是以容易得到的密集組件逼近並覆蓋局部最佳稀疏結構，採用1x1，3x3，5x5卷積核以及max pooling進行不同尺度特徵融合，為了減少計算量再加上1x1的卷積核進行降維。

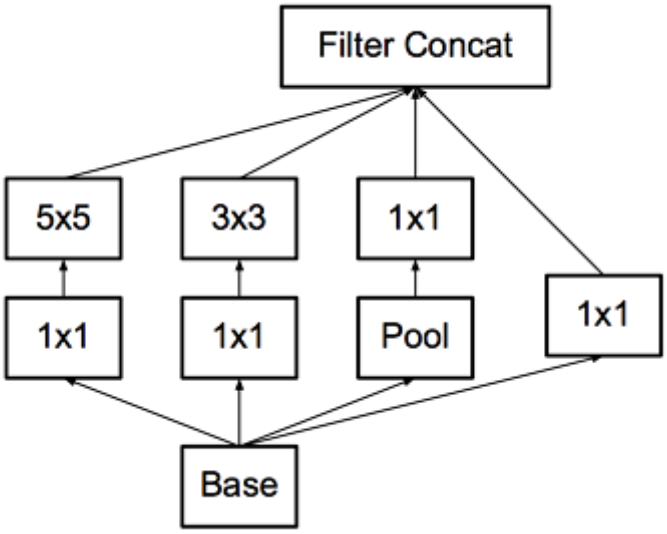

圖5： Original Inception module

Inception V3改良Inception 模塊，將大尺寸的濾波器分解到更小的卷積減少計算複雜度，例如將5x5卷積拆解成兩個3x3卷積或是將7x7卷積拆成1x7與7x1卷積，以及引入輔助分類器達到更好更穩定的收斂，縮小網格的同時擴展Inception模塊。 Inception V3在ILSVRC 2014的Top-1錯誤率最佳結果為4.2%，幾乎是該項比賽冠軍GoogLeNet的一半。
InceptionResNet V2借鑑了ResNet的殘差連接進一步改進Inception V3，建構了更深更複雜的網路。

Type	Patch size/strides or remarks	Input size
Conv	3x3/2	299x299x3
Conv	3x3/1	149x149x32
Conv padded	3x3/1	147x147x32
pool	3x3/2	147x147x64
Conv	3x3/1	73x73x64
Conv	3x3x2	71x71x80
Conv	3x3x1	35x35x192
Interception	3 Interception modules	35x35x288
Interception	5 Interception modules	17x17x768
Interception	2 Interception modules	8x8x1280
Pool	8x8	8x8x2048
Linear	Logits	1x1x2048
Softmax	classifier	1x1x1000

圖6： Interception v3 architecture

- Xception
  Xception是對Inception V3的另一種改進，採用類似depthwise separable convolution的模組先做1x1卷積，在對1x1卷積的每個channel分別進行3x3卷積操作，將結果concat，這種模組稱為SeparableConvolution。Xception網路結構在操作之間以ReLU激活，並結合了ResNet的殘差連接,與InceptionV3相比在準確率與收斂速度上都有進一步提升。

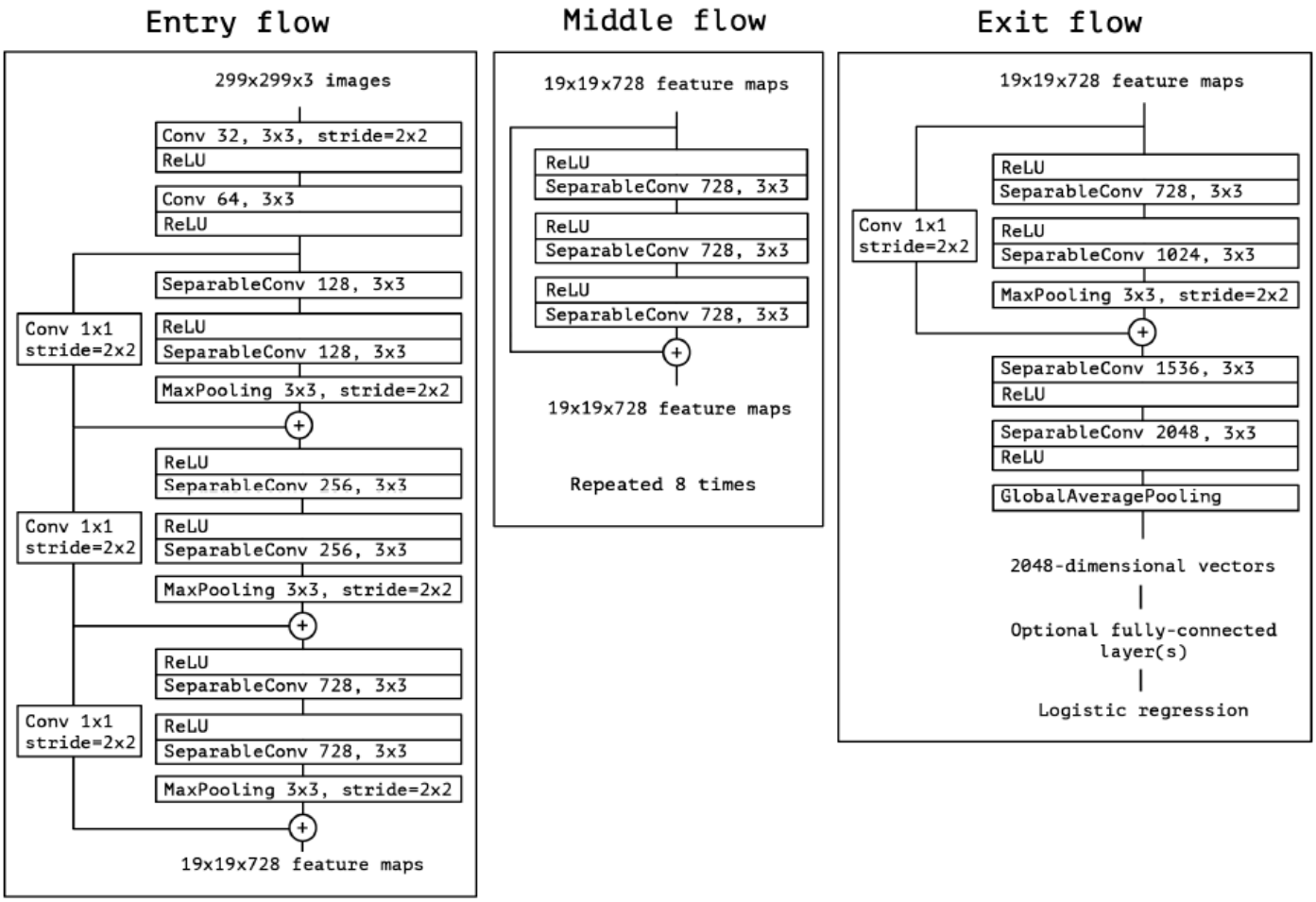

圖7： Xception architecture

- DenseNet

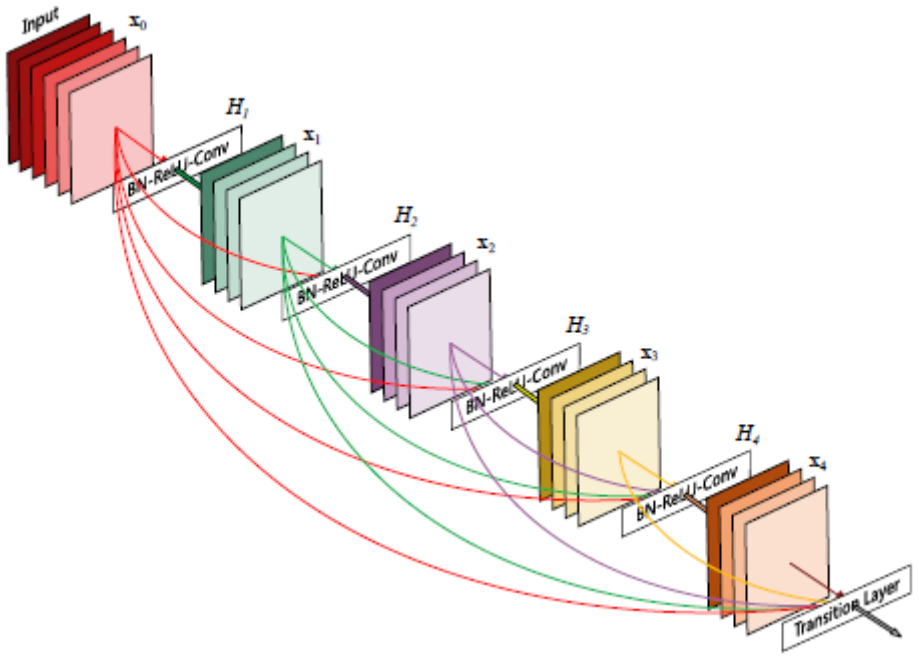

圖8： DenseNet

DenseNet提出了密集連接卷積神經網路，以前饋的方式將每個層與其他的層連接起來，因此一個具有L層的網路架構將有$\frac{L(L+1)}{2}$個連接。DenseNet可以減緩梯度消失、加強特徵傳播、特徵重用，大大減少了參數數量。在一個Dense Block中，令x_l為第l層的輸入，$H_l(\cdot)$為DenseNet的複合函數，則$x_l = H_l([x_0, x_1, \ldots, x_l])$。$H_l(\cdot)$會先進行批量歸一化，接續一個ReLU單元，最後是一個3x3的卷積。Dense Block與Dense Block之間有進行批量歸一化、1x1卷積和2x2平均池化的過度層。DenseNet是CVPR 2017的最佳論文。

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

圖9：DenseNet architecture

- Adam
  Adam名稱的由來是adaptive moment estimation，是一種對隨機目標函數執行一階梯度優化的算法。Adam算法結合了Momentum和RMSProp的優點並加以改進，Momentum通過對梯度均值的估計減少震盪加快學習率，RMSProp則是用梯度除以二階矩陣的平方根作為梯度的下降速度。由於m與v初始值為0，因此會向0偏置，Adam對偏差做了校正並以超參數beta1和beta2控制移動均值的衰減率，能更好的自動適應學習率，加快收斂。簡化版的算法如下（代碼參考CS231n[11]）：

```
m = beta1*m + (1-beta1)*dx
v = beta2*v + (1-beta2)*(dx**2)
x += - learning_rate * m / (np.sqrt(v) + eps)
```

基准模型

我選擇當作參考的基準模型是帶有預訓練的VGG-16。根據Kaggle網站本次比賽論壇的討論，該模型能夠達到0.23800的log loss，在排名榜上足以排名前9%。我會將我的模型與此模型進行比較，測試是否可以達到更好的log loss，並嘗試使用混合模型或超參數調優來進一步提高分數。

III. 方法

数据预处理

在第二部分的數據探索中提到了需要依照司機的ID劃分訓練集與驗證集，我試用了兩種方法，首先用Pandas載入標記司機ID與圖像的CSV檔，取得每個圖像的司機ID、檔案路徑與對應的標籤，接著隨機劃分訓練集與驗證集的司機ID。第一種方法分別從訓練集與驗證集圖像的檔案路徑讀取圖像，將格式轉成uint8節省內存消耗，產生訓練集與驗證集的numpy array使用fit函數輸入給模型。第二種方法依據訓練集與驗證集圖像的檔案路徑在一個新的資料夾產生符號連結，用ImageDataGenerator建構訓練集與驗證集的生成器，使用fit_generator函數輸入給模型。我將每種模型的預處理函數加入模型的Lambda層，利用GPU進行預處理進一步節省時間。

第一種方法執行較快，但是比第二種方法更消耗內存，使用第二種方法的另一個目的是想對圖像做數據增強，但是在之後的實驗顯示對於這項比賽的圖像做數據增強效果不佳，原因是車內的攝影機角度基本上是固定的，對圖像做位移、旋轉、翻轉等反而偏離了目標情境，而且位移圖像也有切到司機的手或頭等重要特徵的風險。

执行过程

這個項目我使用自己的桌上型電腦完成，搭載的GPU是GTX 1080Ti。在建構模型上我首先參考教學[7]的教學先搭載預訓練模型並將模型的include_top設置為False表示不使用模型定義的全連接層，導出預訓練模型的特徵向量，載入特徵向量並定義自己的全連接層開始訓練模型，optimizer測試過SGD與Adam，學習率調整過1e-2到1e-5之間的數值。在這階段我測試了VGG16、ResNet50、InterceptionV3、Xception效果都不佳，訓練20個epoch的log loss還在0.5左右，驗證集的log loss則在1左右。搭載預訓練的特徵向量只訓練後面的全連接層雖然訓練十分快速，但是走神司機的數據集與ImageNet的數據集差異實在太大，唯有放開前面的層訓練才能達到更好的表現。

之後我搭建函數式模型，由上而下的結構為Input層、Lambda層(預處理函數)、選用的模型、GlobalAveragePooling層、Dropout層，最後是以softmax函數激活輸出10種分類預測結果的Dense層。

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 299, 299, 3)       0
_________________________________________________________________
lambda_1 (Lambda)            (None, 299, 299, 3)       0
_________________________________________________________________
xception (Model)             (None, 10, 10, 2048)      20861480
_________________________________________________________________
global_average_pooling2d_1 ( (None, 2048)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                20490
=================================================================
Total params: 20,881,970
Trainable params: 20,827,442
Non-trainable params: 54,528
_________________________________________________________________
```

圖9： 模型結構(以搭建Xception為例)

在訓練模型的過程中常碰到的問題是ResourceExhaustedError，原因是在訓練較複雜的模型時Out of memory。這時候我採用的方法是用ImageDataGenerator載入訓練數據並減少batch size，Keras默認的batch size是32，我使用的GPU內存為11G，在訓練VGG與ResNet以外的模型時調整batch size為16，在嘗試模型融合時要在將batch size減少到8。

完善

決定好架構以後開始選擇要使用的模型，由於目標分數是0.23800，我採取的策略是先調出在驗證集上分數在0.3以下的模型，之後再根據測試集的預測結果提交道Kaggle的分數找出表現較好的幾個模型融合。

首先我使用了VGG16與VGG19，考量到一般情況前面的層只會判斷線條之類的基本特徵，我測試了開放全部層與鎖住前6層的差異，鎖住層的架構驗證集的loss只降到0.45左右，開放全部層的表現明顯有提升，所以之後搭建的模型我都開放了全部的層。調整的參數有Dropout的比例與使用的optimizer和學習率，也測試調整對Dense層做隨機初始化的標準差，在VGG16上使用SGD，學習率1e-3，Dropout 0.6或0.7，Dense層初始標準差0.01可以在驗證集上取得0.29左右的成績，然而提交到Kaggle上測試集的loss在0.8以上，VGG19表現更差，因此我捨棄了這兩個模型。

使用ResNet50調整過Optimizer、學習率、Dropout的比例在驗證集上的loss都還在0.4左右，因此也捨棄了這個模型。

使用Inception V3，Dropout比例測試0.5~0.9的數字，Optimizer採用Adam測試學習率1e-3~1e-5，最後設定Dropout為0.7，學習率1e-4表現最佳，在驗證集上的表現到達0.3的成績，提測試集結果分數0.45。由於log loss算法對預測錯誤的樣本預測值為0或0.005的差距很大，這裡我一併測試對預測結果作clip的效果，後來決定將預測值限制到[0.001, 0.999]的範圍，分數也進步到0.41左右。

Xception是我跑的所有模型中表現最好的，使用Adam的學習率設為1e-4，Dropout比例0.6或0.7，驗證集上達到的loss都在0.3以下，最好的結果為0.23，使用兩種設定的Xception做完clip以後在測試集上的分數都達到了0.36，於是我保留兩種結果。

InceptionResNet V2與前面幾個模型相比參數更多，花費更多的硬體資源，執行時間也更久，在InceptionResNet V2上同樣調整Optimizer、學習率和Dropout的比例，最後使用Dropout 0.8與Adam學習率1e-4的設定，在Kaggle上得到的成績為0.363。

對於DenseNet結構我測試了DenseNet121、DenseNet169與DenseNet201，同樣使用Adam學習率1e-4，Dropout比例0.7的情況下DenseNet169的表現比其他兩個更好，提交Kaggle的分數DenseNet169為0.46，DenseNet201則高達0.84，所以我只保留DenseNet169的結果使用。

由於Xception與InceptionResNet V2是我測試表現最好的模型，我也用函數嘗試搭建混合Xception與InceptionResNet V2的模型，但是執行時間太長，初步嘗試提交混合模型的預測分數只有0.74471，而且對於配置要求太高，想搭建Xception和InceptionResNet V2再加上Inception V3的模型再次耗盡資源，因此最終使用的方案我將多個模型分別做出預測在取平均整合最終結果。

單個模型在測試集上最佳的表現為0.36030，與訂下的目標相去甚遠，於是我開始整合多個模型的預測結果，首先整合Xception、Inception V3、InceptionResNet V2，DenseNet169的結果，loss降到0.27259，

接著再訓練幾個表現較好的模型Xception、Inception V3、InceptionResNet V2實驗融合更多模型的結果，增加的Xception對表現提升效果最好，接著測試剔除DenseNet的結果發現分數也有提升，於是放棄使用DenseNet模型。實驗將分數最高與次高的Xception模型加權2倍，也實驗了加入融合Xception與InceptionResNet V2模型的預測結果，分數都沒有提升。之後對InceptionResNet V2調參加入表現更好的模型，loss降到了0.23932。最後再訓練了使用不同測試集的Xception模型，使用4個Xception，2個InceptionResNet V2，1個Inception V3融合的結果，loss降到0.23790，達到目標。

IV. 结果

模型的评价与验证

下表紀錄了融合模型預測提交到Kaggle上的分數，其中D代表DenseNet169，X代表Xception，I代表Inception V3，IR代表InceptionResNet V2，中間有乘法代表對其中一個模型的結果作加權處裡，XI則是融合Xception與InceptionResNet V2的模型。

Model	**Score**
D+X+I+IR	0.27259
D+X+X+I+IR	0.26563
D+X+X+I+I+IR	0.26104
D+X+X+X+I+I+IR	0.24928
X+X+X+I+I+IR	0.24645
X+X+X+I+IR	0.24203
X+X+X+I+IR+IR	0.23945
X+X+X+IR+IR	0.26260
XI+X+X+I+IR+IR	0.25561
X+X*2+I+IR+IR	0.24290
X*2+X+I+IR+IR	0.24113
X+X+I+IR	0.24634
X+X+X+I+IR+IR	0.23932
X+X+X+X+I+IR+IR	0.23790

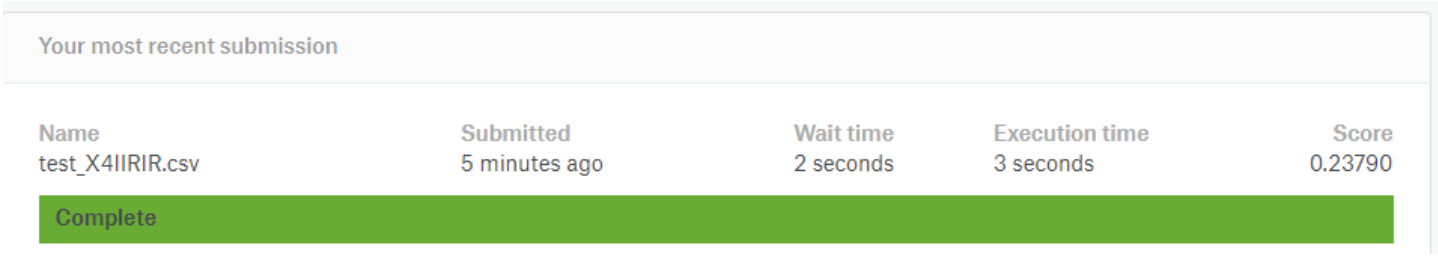

圖10：最終得分

合理性分析

得到了高於基準模型目標log loss以後，我們來觀察其中一個使用的模型對驗證集預測的熱圖。在驗證集中表現最差的是talking to passenger分類，觀察此類別分類錯誤司機大多是雙手還握在方向盤上，臉朝向副駕駛座的人，被模型判斷為安全駕駛。這裡主要能判斷類別的資訊可能只有嘴型，因為臉的方向也常常跟路況有關，雙手同樣都握著方向盤，可以理解為模型未充分學習利用嘴部或是眼睛的特定視角做為特徵判斷。另外數據集對這些圖像的分類，以及正常駕駛與跟乘客交談的數據中有多少這類型的數據都會影響到結果。

另外一個常見的分類錯誤是將talking to passenger左或是右的分類誤判為hair and makeup，這部分較可能的原因是模型沒有檢測出圖像中的手機，同樣的問題在喝飲料的分類中也有機會出現。

模型對於未涉及物品檢測或是如texting這種有明顯特徵的分類上都有不錯的結果，在較困難的判斷透過模型融合也能達到更好的結果

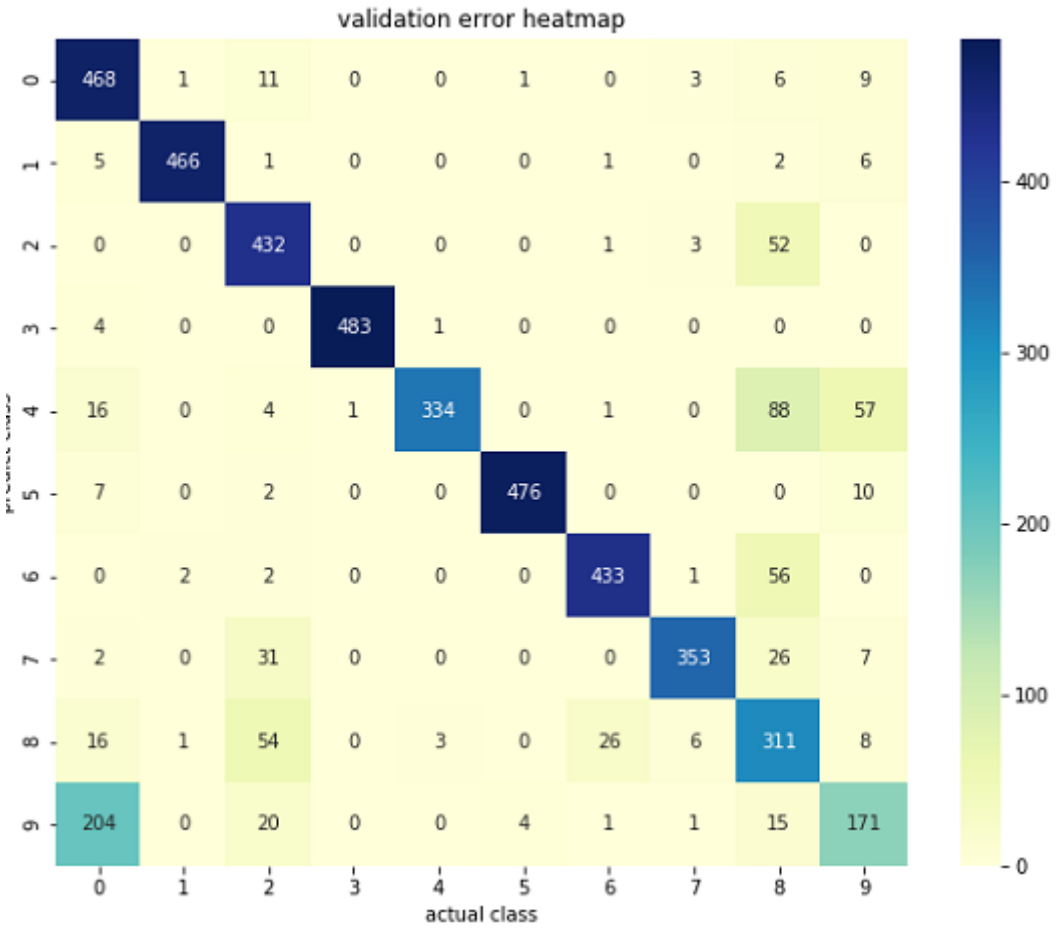

圖11：其中一個Xception對驗證集預測的熱圖

圖12：分類錯誤的talking to passenger

V. 项目结论

结果可视化

在這個項目中我對數據進行了Class Activation Mapping(CAM)[12]處理來將特徵可視化。

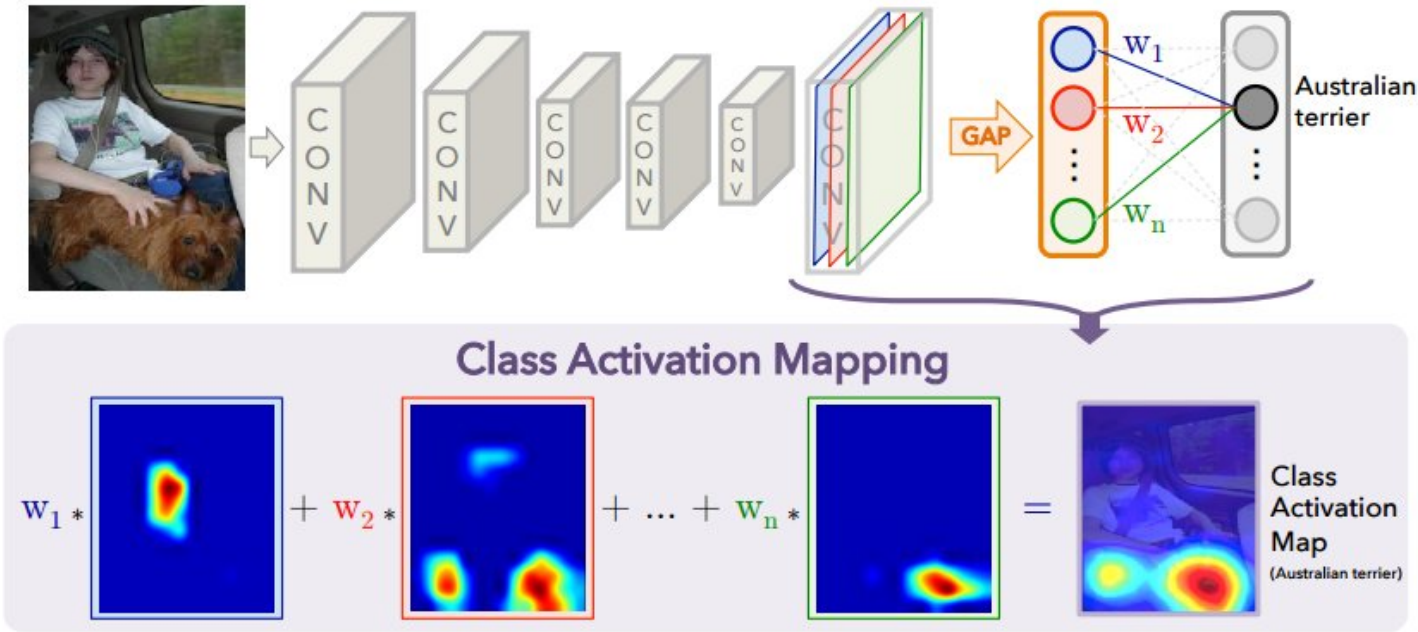

圖13：Class Activation Mapping

CAM主要是通過Global Average Pooling(GAP)來實現，GAP通常介於最後一個卷積層與softmax層中間，主要作用是取代全連接層的複雜計算並且對參數進行正則化。假設$f_k(x,y)$代表第K個feature map上(x,y)位置的值，GAP層的結果可以表示成$F_k=\sum_{x,y}f_k(x,y)$，對於某個類別c，softmax層的輸入值為$s_c=\sum_k w_k^c F_k$，輸出的預測值為$P_C=\frac{exp(s_c)}{\sum_c exp(s_c)}$,將前面的公式展開可以得到：

$$Sc = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y)$$

定義M_c為某一個類別c的Class Activation Map， 其中對應位置(x,y)的元素為：

$$M_c(x,y) = \sum_k w_k^c f_k(x,y)$$

則$S_c = \sum M_c(x,y)$，而$M_c(x,y)$表示了位置(x,y)對於激活函數的重要性，可以由$M_c(x,y)$生成類別c的CAM圖像。

從測試集隨機取9張圖預測得到的CAM圖如下：

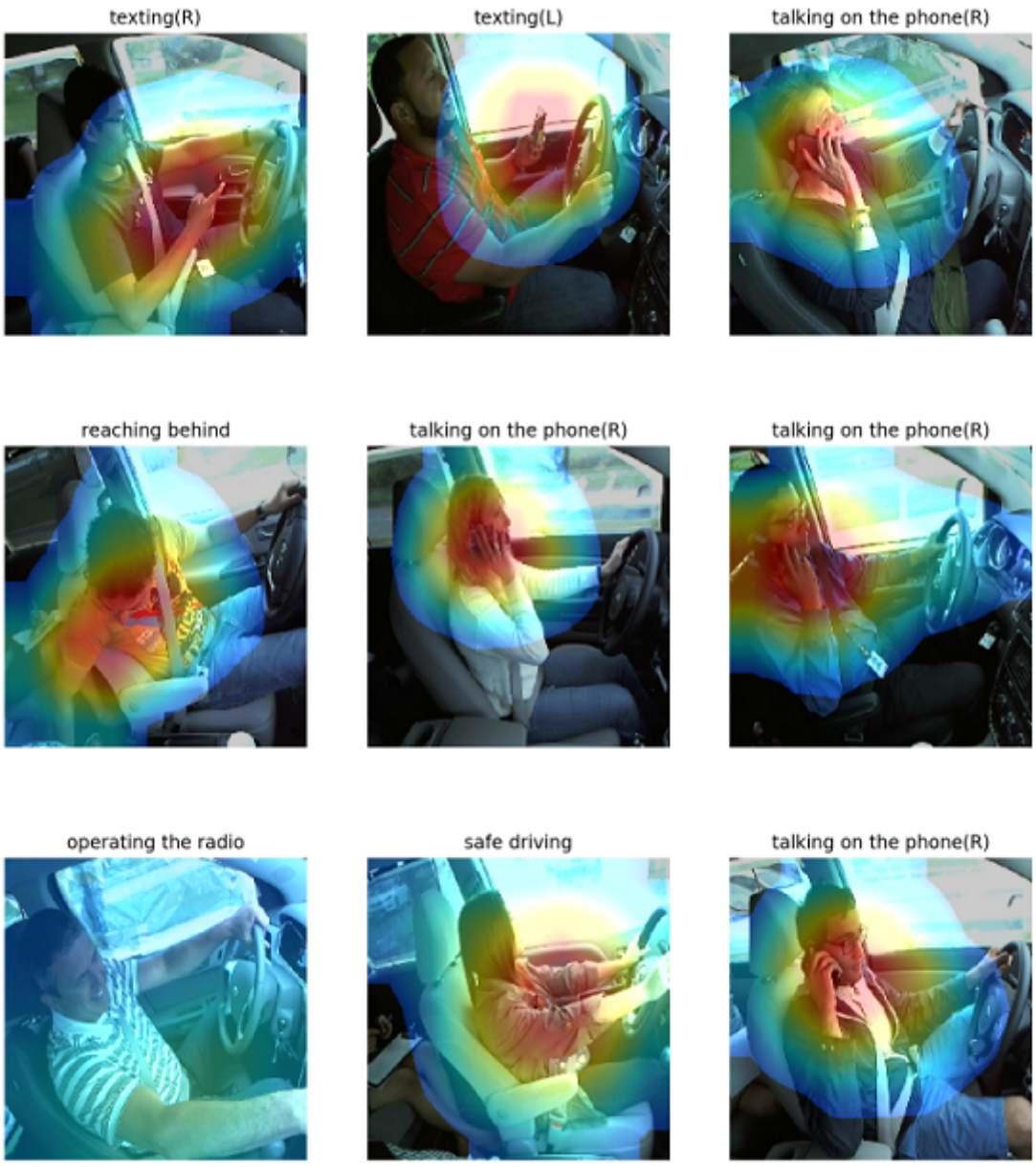

圖14： Randam CAM 9 drivers

從CAM的熱點可以看到模型分類出發短信、打電話的關鍵特徵在於圖像中有無手機，拿手機的手是左手或右手，手機是拿在耳朵旁邊還是拿在手上。對於沒有手機的圖模型關注在身體方向，兩手的位

置，即使畫面沒有出現收音機也能夠判斷手的位置是否在接近收音機的方向，另外對司機視線不在正前方的圖也根據兩手皆放在方向盤上推測出安全駕駛。此一結果與我預期看到的情況相符，因此我認為模型有正確學習到分辨這些不同狀態需要觀察的特徵。

对项目的思考

這個項目中我首先分析了問題的情境與嚴重性，構建一個使用卷積神經網路的解決方案並訂下一個明確的目標分數。對數據集先找出需要注意的相似性，然後測試幾個著名的模型架構並測試鎖住層與否的模型表現，調整模型的參數並用CAM處裡確認模型學到特徵的合理性，最後整合其中表現較好的模型預測成為最終的預測結果。

我覺得這個項目有趣的地方是可以實際解決生活中面臨到的需要被解決的問題，為幫助社會進步盡一份心力。困難的地方是應該是數據集的特性，與之前做過的狗分類不同，這個項目的數據集有太多相似的圖像，很容易發生過擬合，Optimizer使用Adam試了許多不同組合模型的loss大部分在前5個epoch以內大致底定，之後就呈現震盪的狀態。訓練集只有測試集的1/3左右，與一般理想情況訓練集與測試集大約8:2左右的劃分相距甚遠，或許這是比賽項目常會碰到的情況，另外數據增強在這個項目難以發揮較果也限制了進一步優化的空間。

最終的預測結果整合了7個模型的輸出，在Kaggle上達到比基準模型更優的log loss，從特徵的可視化的確認了模型學到了合理的特徵，因此我認為最終得到的結果符合期望。如同這個比賽的目的，我認為在通用場景下將這個模型應用在儀表板相機產品，可以在司機未專注駕駛的時候判斷司機的行為並提示出警告訊息。

需要作出的改进

我認為首先提升硬體效能是需要努力的方向，為了建構更複雜的模型並加快訓練速度與調參效率。這次受限於硬體沒有太多模型融合的空間，用多筆預測平均得到最終結果是我覺得較可惜的地方，我希望之後能搭建包含三個以上模型架構或者用集成方法整合不同模型。另外為了簡化驗證過程我在這個項目未使用K-Fold交叉驗證確認模型表現，為了要更精準的評估模型效能這也是需要改進的地方。短期之內我會先學習aws的設定用p3.2xlarge環境搭建模型，之後再考慮組裝搭載多張GPU的桌上型電腦。

觀察模型的熱圖我也觀察到模型做物品檢測也有加強空間，可以考慮結合一個專門檢查手機和飲料杯的模型確保能在數據集中找到這些重要特徵。

另一個可以做的改進是數據清洗，我在此項目中大致檢查過沒有異常的圖片存在，然而還是有樣本分類錯誤的可能性。可以嘗試整理出模型在訓練集中分類錯誤的樣本，檢查其中是否有分類錯誤樣本。

Reference

[1] CDC Injury Center, Motor Vehicle Safety, from https://www.cdc.gov/motorvehiclesafety/distracted_driving/

[2] Kaggle, State Farm Distracted Driver Detection, from https://www.kaggle.com/c/state-farm-distracted-driver-detection

[3] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] HE, Kaiming, et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p. 770-778.
[5]Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
[6] CHOLLET, François. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, 2016.
[7] SZEGEDY, Christian, et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *AAAI*. 2017. p. 12.
[8]Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. Vol. 1. No. 2. 2017.
[9]Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
[10]Karpathy, Andrej. "Cs231n convolutional neural networks for visual recognition." *Neural networks* 1 (2016).
[11]Yang Peiwen, 手把手教你如何在Kaggle猫狗大战冲到Top2%, from https://ypw.io/dogs-vs-cats-2/
[12] ZHOU, Bolei, et al. Learning deep features for discriminative localization. In: *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016. p. 2921-2929.