CNN fruit level classification

data pre-procesing

(1) rescale pixel value
將pixel value 做標準化到0和1之間,減少需要的運算資源,避免gradient explosion
(2) data augmentation

zoom_range	height_shift_range	width_shift_range
[0.7 1.3],隨機縮放0.7倍～1.3倍	20%, 圖片垂直平移量為高度的20%	20%, 圖片左右平移量為寬度的20% 1.3倍

Color Jitter:
由於芒果照片的拍攝地點及環境光亮度不是固定的,但不同環境下拍攝照片的比例不太一樣,為了減少環境影響,所以將圖片的亮度、對比、銳度、顏色做一些隨機變化,讓model比較能適應不同環境,但顏色也是影響芒果等級的因素之一,所以不能有太誇張的變化,變化範圍都不超過-5%~+5%

Machine learning approaches

(1) cnn model with inception module

dataflow:input->part1->part1->part2->part2->part3->part3->part4->output
part1:
這個cnn model是參考[1],將圖片用兩種大小不同的mask做convolution,分別為3x3、7x7,其中跟3x3 mask做3次convolution後做3x3 maxpooling,再跟7x7 mask做convolution,最後再做一次3x3 maxpooling,用7x1、1x7兩種mask做asymmetric convolution代替7x7 mask以減少參數量,一開始的做3次3x3 convolution的目的是要以多次用較小mask做convolution的方式取代一個較大的mask,但不使用上述asymmetric convolution的原因是論文中題到在asymmetric convolution在lower layer的效果比較差[2]。
part2:
設計4個branch[3],每個branch分別做不同scale的convolution
branch1:average pooling->1x1 convolution
branch2:1x1 convolution
branch3:1x1 convolution->3x3 convolution
branch4:1x1 convolution->3x3 convolution->3x3 convolution(取代用較大的mask做convolution)
將part1得到的frature map分別送進4個branch,再把4個branch的output合併成一個channel更深的output

part3:
將part2的output當成input
將part2中的所有3x3 convolution以7x1、1x7 asymmetric convolution取代(7x1、1x7 asymmetric convolution是用來取代直接用較大的mask,並減少訓練參數)
把4個branch的output合併成一個channel更深的output
part4:
對part3的output做average pooling在接兩層fully connected layer,輸出neuron數量分別為1024,3(類別數量),在第一層fully connected後做dropout

parameters and result:

learning rate	Dropout	epoch	val_loss	val_acc	activation	training_acc
1e-3(實驗結果最佳)	0.2	30	0.51	0.78	leaky relu:0.11	0.81

Disscussion
設計4個branch再將output合併的目的是想透過結合不同大小的mask找到optimal local construction,這相當於用不同scale看相同圖片,用neural network學習如何利用這些不同scale的結果,正確判斷物品種類
每個branch 1x1 convolution的功能:
上一層output的channel數量增加太多,為了計算效率,利用1x1 convolution降維,將上一層的output的channel壓縮到更少的channel。
為了減緩overfit,利用dropout,earltstop及modelcheckpoint的方式找出最好的model。
layer越深spatial concentration越低,所以我們讓比較深的layer有比較多mask。training graph link (https://imgur.com/Oe2UpkX)

(2) Integrated stacking ensemble

submodel1:
訓練keras提供的Inception-v3 pretrained model,這個model的訓練資料種類分常廣,可以分出1000類物品,所以我們希望透過Inception-v3 pretrained model抽取feature的能力加上一些fine-tune之後,可以對stacked model有幫助,如此可以省下訓練時間及解決訓練資料不夠的問題。
training graph link (https://imgur.com/uTvnKbH)
submodel2:
因為A,B等級芒果的特徵差距不明顯,approach1最容易分辨錯誤的也是A,B等級,所以我們train了一個比較深度比較深的cnn當做feature extractor配合svm,不使用submodel1當extractor除了訓練時間太久之外,對於差別太小卻不同等級的分類效果不算好,所以我們想做一個比較深,專門提取芒果表皮特徵的extractor。extractor training graph link (https://imgur.com/3OcTUiv)
stacking:
將approach1,submodel1,submodel2的預測結果接在一起變成一個9維的向量,當作stacked model的input。
為了減緩overfit,訓練stacked model用的training data是網站提供的validation data。

stacked model本身是一個簡單的large multi-headed neural network,除了submodels之外,多加兩層fully connected layer,主要功能是可以結合所有submodels的預測結果,利用neural network學習如何以最好的方式利用這些結果,在進行訓練時,不對submodels任何一個layer的權重做更新。training graph link (https://imgur.com/8dXJRWj)

parameters and result:

learning rate	Dropout	activation	epoch	val_loss	val_acc	train_acc
1e-3(實驗結果最佳)	0.4	leaky relu:0.11	30	0.388	0.856	0.858

Disscussion
可以發現performance有明顯變好。
其實一開始在訓練stacked model時是用training data+一半的validation data,整個訓練的過程的training accuracy都很好,也比submodels高很多,但是用剩下的一半validation data做testing之後,結果比submodels還慘,後來我們猜測是因為training data太多導致stacked model只是在幫submodels繼續訓練,而不是學著利用submodels的結果,所以最後決定只用一點做過color jitter的training data及一半validation data做訓練,另一部份做testing,結果發現training accuracy跟validation accuracy差距縮小很多。

(3) Extreme Gradient Boosting(xgboost)

利用submodel2提到的frature extractor取出frature當成input。
xgboost是Additive Training,每一次保留原來的模型不變,並且加入一個新的tree function至model中(如圖link (https://imgur.com/bFMxHvV)),修正上一棵樹的錯誤。在更新時對loss function進行Taylor Expansion,目的是為了優化loss functionu,也就是對第t個樹結構做最好的分裂。
xgboost在loss function中做了regression,可以控制模型的複雜度,降低了model的variance,這相當於做了pruning,可加快訓練速度。
parameters and result:

learning rate	booster	max_depth	loss	accuracy	# estimators
0.05	Decision Tree	20	0.399	0.80125	200

comparison

efficiency:
以準確度來看stacked model是目前表現最好的,但容易overfit(除非很小心使用),訓練時間最短的則是xgboost,不過我們這組還是以準確度為優先考量
scalability:
approach1的model相對於stacked model(approach2)的計算複雜度比較低,占用的計算資源比較

少。xgboost需要一次載入資料,這是package限制,所以資料越多或是圖片解析度越高,xgboost需要的記憶體大小會增加非常快,approach1則是只需要處理一個batch的記憶體,再加上asymmetric convolution可以減少訓練參數量,所以需要的記憶體資源比較低;只要資料量太多無法一次載入,xgboost必需採用incremental learning的方式一次訓練一部份的資料,如果某部份資料分布不均勻,會影響model的正確性,讓performance變差,所以scalability最好的是approach1

popularity:

xgboost再提升model performance上應該是比較有名的也常常在一些比賽中出現,approach2的model是結合最近幾年比較受歡迎的其中一個CNN model以及model stacking的技巧,如果以影像辨識來說,應該是approach2的方法比較常被用到

interpretability:

我們這組覺得approach2的interpretability最佳,因為我們認為結合不同大小的mask可以找到optimal local construction,可以取出更具有代表性的frature,這相當於用不同scale看相同圖片;再加上結合model stacking的技巧,可以減輕單一個model出錯造成的影響,對提升performance很有幫助

recommend:

approach2

cons and pros of the recommended model

approach2訓練起來很方便而且訓練時間很短(不包含submodels的訓練時間),只要有一些已經訓練好而且表現不差的model就可以利用這些submodels訓練出一個variance和bias都比較低的model,以final project來說,model stacking可以很有效的提升準確率。

approach2的壞處就是太容易overfit,如果submodels本身的複雜度已經很高了,stacked model很容易就overfit。另外,在幾次的實驗中發現,如果其中一個submodel有嚴重的overfit或是表現太差會讓stacked model學不到東西,我們猜測這是因為我們只用3個submodels,所以一個太差的submodel會造成stacked model input的noise比例太高。

work loads

陳麒宇:主要負責一開始的data pre-processing、approach2的submodel2
張皓鈞:主要負責approach3、approach2的submodel1
何瑞祥:主要負責approach1及approach2的model stacking
之後修改model都是跟組員討論後再修改自己負責的部份

Competition result

reference

[1]Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9 (https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html)

[2]Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826 (https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html)

[3]SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A... Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. AAAI Conference on Artificial Intelligence, North America, feb. 2017. (https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806/14311)