

## 實戰| OpenCV實現棋盤圖像識別

計算機視覺life 前天

以下文章來源於小白學視覺，作者努比



**小白學視覺**

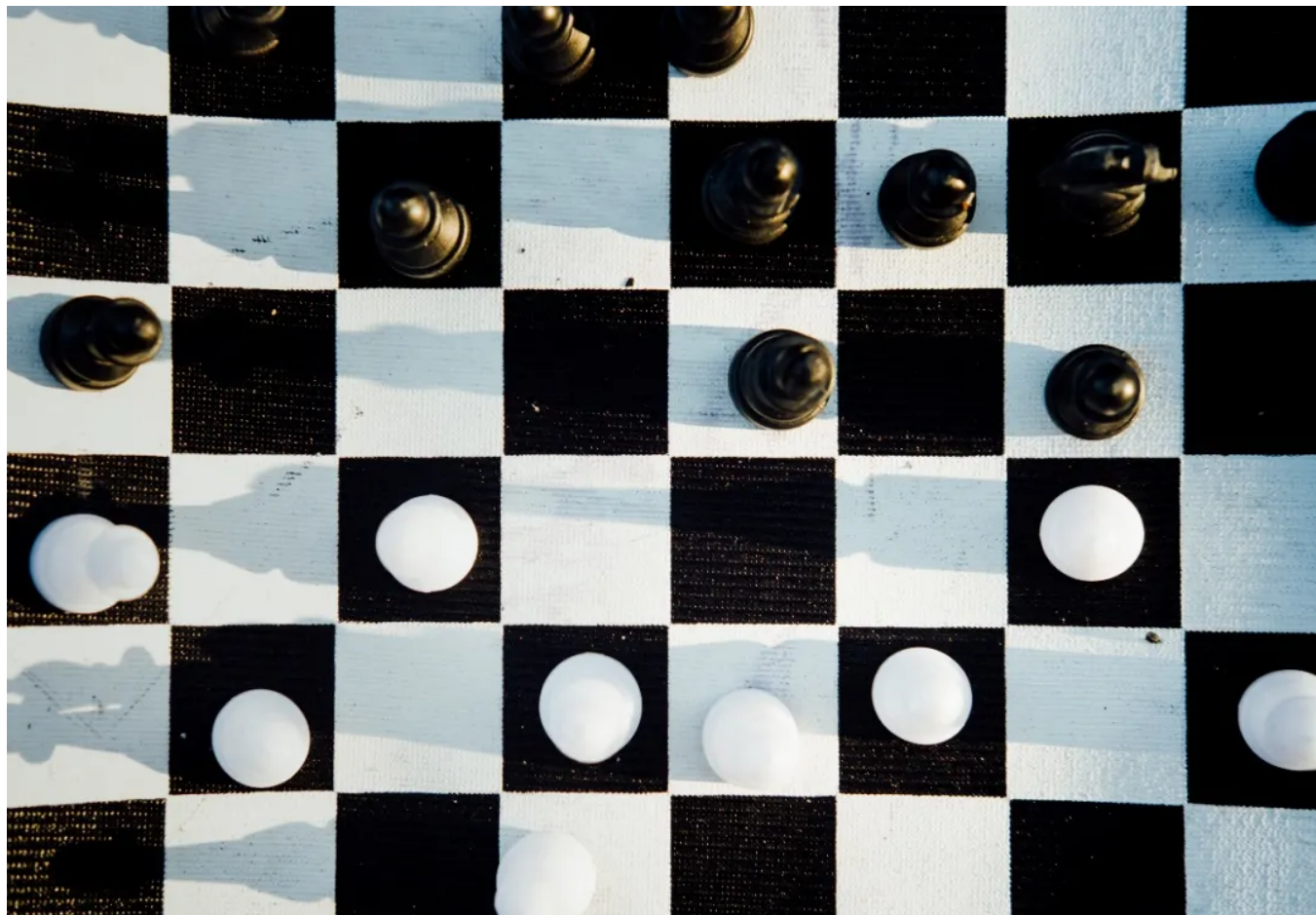
哈工大在讀博士的公眾號，《OpenCV 4快速入門》的作者，面向初學者介紹計算機視...



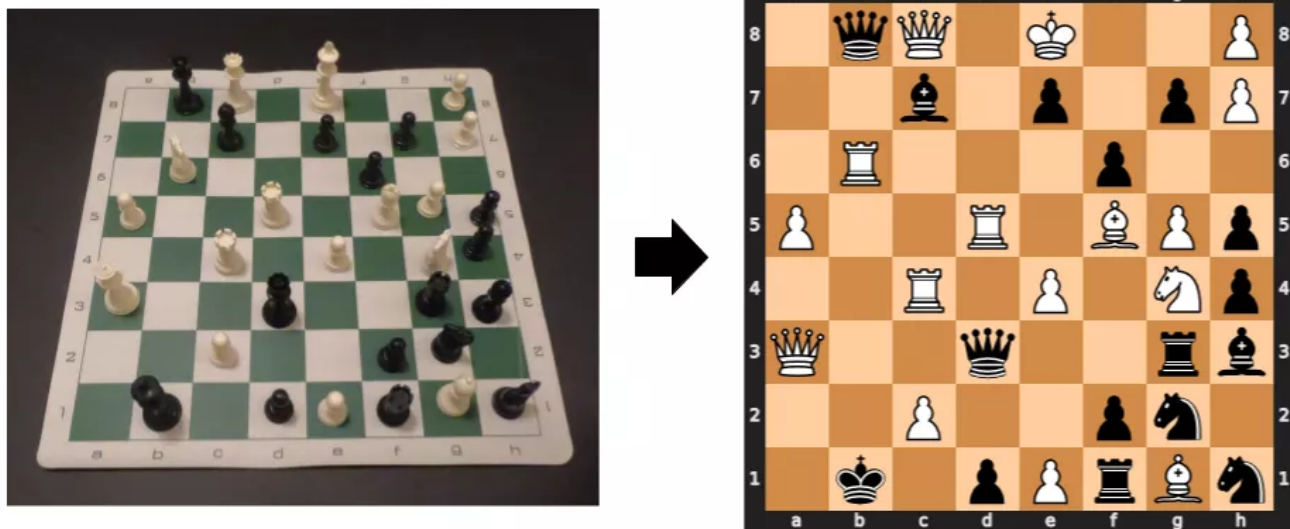
點擊上方“[計算機視覺life](#)”，選擇“星標”

快速獲得最新乾貨

本期我們將一起學習如何使用計算機視覺技術識別棋子及其在棋盤上的位置



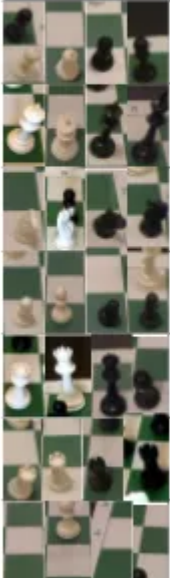
我們利用計算機視覺技術和卷積神經網絡（CNN）為這個項目創建分類算法，並確定棋子在棋盤上的位置。最終的應用程序會保存整個圖像並可視化的表現出來，同時輸出棋盤的2D圖像以查看結果。



(左) 實時攝像機進給的幀和棋盤的 (右) 二維圖像

## 01. 數據

我們對該項目的數據集有很高的要求，因為它最終會影響我們的實驗結果。我們在網上能找到的國際象棋數據集是使用不同的國際象棋集、不同的攝影機拍攝得到的，這導致我們創建了自己的數據集。我使用國際象棋和攝像機（GoPro Hero6 Black以“第一人稱視角”角度）生成了自定義數據集，這使我的模型更加精確。該數據集包含2406張圖像，分為13類（請參閱下文）。**總結：這花費了我們很多時間，但是這使得訓練圖像盡可能地接近在應用程序中使用時所看到的圖像。**

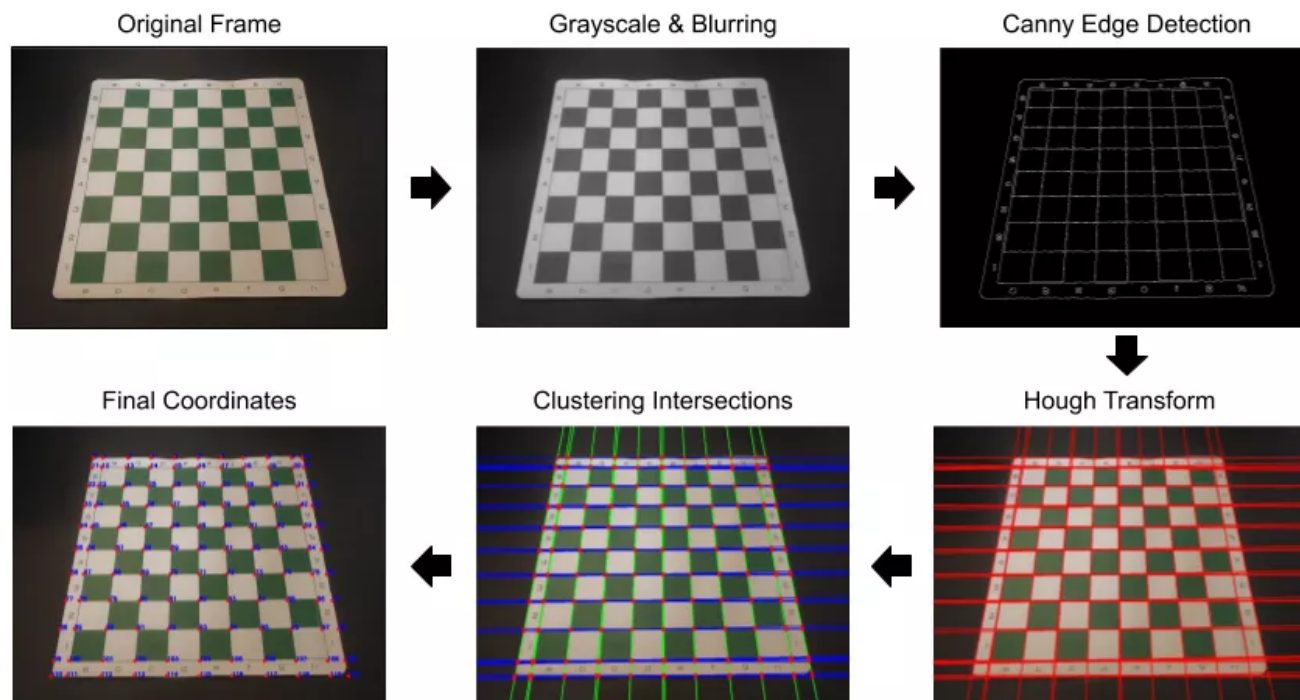
Category		White	Black
	Bishop	216	201
	King	111	104
	Knight	206	172
	Pawn	200	195
	Queen	199	186
	Rook	200	199
	Empty	199	

自定義數據集的細分

為了構建該數據集，我首先創建了capture\_data.py，當單擊S鍵時，該視頻從視頻流中獲取一幀並將其保存。這個程序使我能夠無縫地更改棋盤上的棋子並一遍又一遍地捕獲棋盤的圖像，直到我建立了大量不同的棋盤配置為止。接下來，我創建了create\_data.py，以使用下一部分中討論的檢測技術將其裁剪為單獨小塊。最後，我通過將裁剪後的圖像分成帶標籤的文件夾來對它們進行分類。

## 02. 棋盤檢測

對於棋盤檢測，我想做的事情比使用OpenCV函數findChessboardCorners複雜的多，但又不像CNN那樣高級。使用低級和中級計算機視覺技術來查找棋盤的特徵，然後將這些特徵轉換為外邊界和64個獨立正方形的坐標。該過程以Canny邊緣檢測和Hough變換生成的相交水平線、垂直線的交點為中心。層次聚類用於按距離對交叉點進行分組，並對各組取平均值以創建最終坐標（請參見下文）。



完整的棋盤檢測過程

### 03. 棋盤分類

項目伊始，我們想使用Keras / TensorFlow創建CNN模型並對棋子進行分類。但是，在創建數據集之後，僅考慮CNN的大小，單靠CNN就無法獲得想要的結果。為了克服這一障礙，我利用了ImageDataGenerator和transfer learning，它增加了我的數據並使用了其他預訓練的模型作為基礎。

創建CNN模型

為了使用GPU，我在雲中創建並訓練了CNN模型，從而大大減少了訓練時間。**快速提示：**  
**Google Colab**是使用GPU快速入門的簡便方法。為了提高數據的有效性，我使用了ImageDataGenerator來擴展原始圖像並將模型暴露給不同版本的數據。

ImageDataGenerator函數針對每個時期隨機旋轉，重新縮放和翻轉（水平）訓練數據，從本質上創建了更多數據。儘管還有更多的轉換選項，但這些轉換選項對該項目最有效。

```
1 from keras.preprocessing.image import ImageDataGenerator
2 datagen = ImageDataGenerator(
3     rotation_range=5,
4     rescale=1./255,
5     horizontal_flip=True,
6     fill_mode='nearest')
7 test_datagen = ImageDataGenerator(rescale=1./255)
8 train_gen = datagen.flow_from_directory(
9     folder + '/train',
10    target_size = image_size,
11    batch_size = batch_size,
12    class_mode = 'categorical',
13    color_mode = 'rgb',
14    shuffle=True)
15 test_gen = test_datagen.flow_from_directory(
16    folder + '/test',
17    target_size = image_size,
18    batch_size = batch_size,
19    class_mode = 'categorical',
20    color_mode = 'rgb',
21    shuffle=False)
```

我們沒有從頭開始訓練模型，而是通過利用預先訓練的模型並添加了使用我的自定義數據集訓練的頂層模型來實現轉移學習。我遵循了典型的轉移學習工作流程：

- 1.從先前訓練的模型（VGG16）中獲取圖層。



```
1 from keras.applications.vgg16 import VGG16
2 model = VGG16(weights='imagenet')
3 model.summary()
```

2.凍結他們，以避免破壞他們在訓練回合中包含的任何信息。

3.在凍結層的頂部添加了新的可訓練層。

```
1 from keras.models import Sequential
2 from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten
3 from keras.models import Model
4 base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224,
5
6 # Freeze convolutional layers from VGG16
7 for layer in base_model.layers:
8     layer.trainable = False
9 # Establish new fully connected block
10 x = base_model.output
11 x = Flatten()(x)
12 x = Dense(500, activation='relu')(x)
13 x = Dense(500, activation='relu')(x)
14 predictions = Dense(13, activation='softmax')(x)
15 # This is the model we will train
16 model = Model(inputs=base_model.input, outputs=predictions)
17 model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=[
```

4.在自定義數據集上訓練新層。

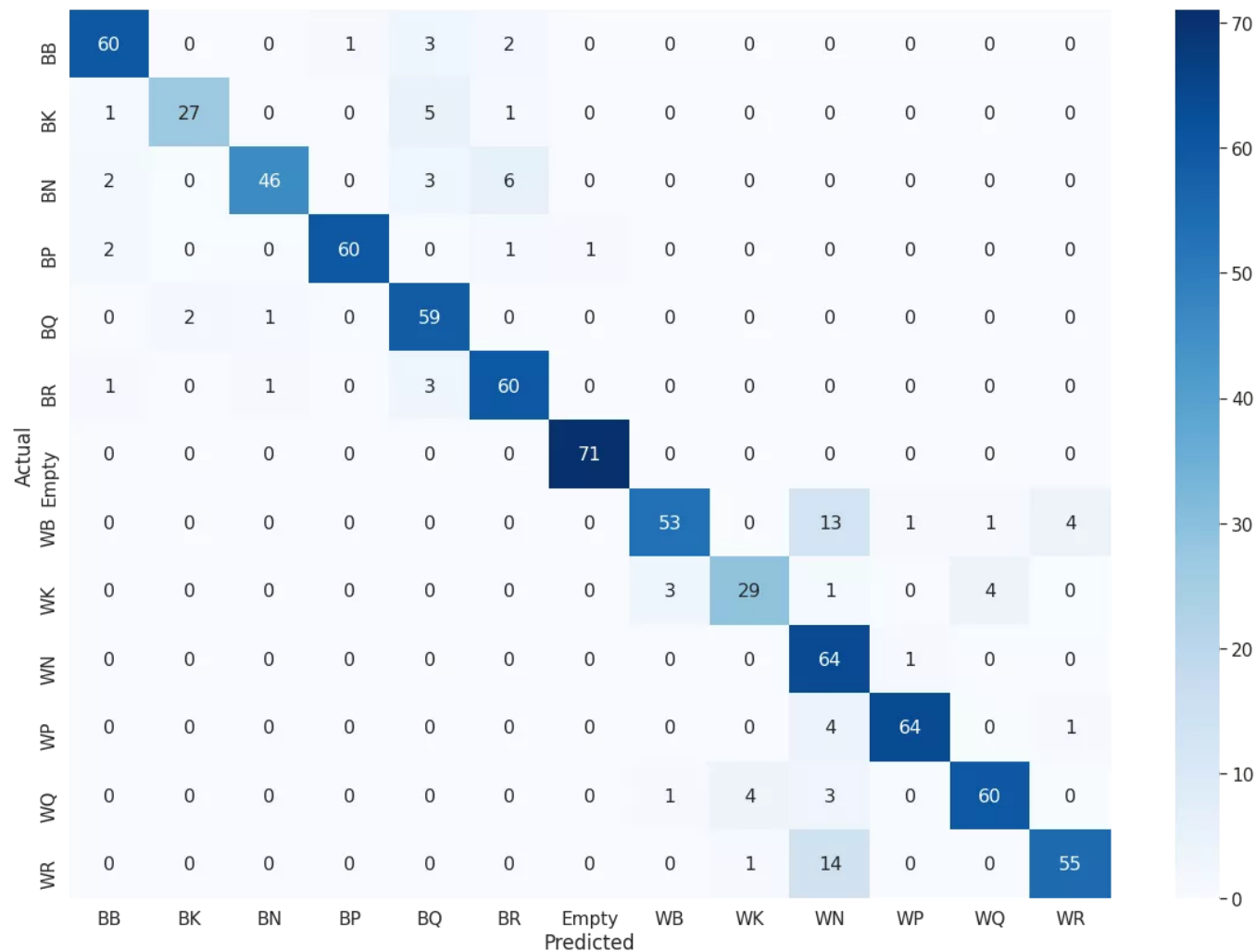
```
1 epochs = 10
2 history = model.fit(
3     train_gen,
4     epochs=epochs,
5     verbose = 1,
6     validation_data=test_gen)
7 model.save_weights('model_VGG16.h5')
```

當我們使用VGG16或VGG19作為預訓練模型創建模型時，由於驗證精度更高，因此選擇了使用VGG16的模型。另外，最佳epochs是10。任何大於10的數均不會使驗證準確性的提高，也不會增加訓練與驗證準確性之間的差異。**總結：轉移學習使我們可以充分利用深度學習在圖像分類中的優勢，而無需大型數據集。**

## 04. 結果

為了更好地可視化驗證準確性，我創建了模型預測的混淆矩陣。通過此圖表，可以輕鬆評估模型的優缺點。**優點：**空-準確率為99%，召回率為100%；白棋和黑棋（WP和BP）-F1得分約為95%。**劣勢：**白騎士（WN）-召回率高（98%），但準確性卻很低（65%）；白主教（WB）-召回率最低，為74%。





測試數據的混淆矩陣

## 05. 應用

該應用程序的目標是使用CNN模型並可視化每個步驟的性能。我們創建了cv\_chess.py，它清楚地顯示了步驟，並創建了cv\_chess\_functions.py，它顯示了每個步驟的詳細信息。此應用

程序保存實時視頻流中的原始幀，每個正方形的64個裁剪圖像以及棋盤的最終2D圖像。

```
1 print('Working...')
2
3 # Save the frame to be analyzed
4 cv2.imwrite('frame.jpeg', frame)
5
6 # Low-level CV techniques (grayscale & blur)
7 img, gray_blur = read_img('frame.jpeg')
8
9 # Canny algorithm
10 edges = canny_edge(gray_blur)
11
12 # Hough Transform
13 lines = hough_line(edges)
14
15 # Separate the lines into vertical and horizontal lines          h_lines, v_
16
17 # Find and cluster the intersecting
18 intersection_points = line_intersections(h_lines, v_lines)      points =
19
20 # Final coordinates of the board
21 points = augment_points(points)
22
23 # Crop the squares of the board and organize into a sorted list          x_lis
24 img_filename_list = grab_cell_files()          img_filename_list.sort(key=na
25
26 # Classify each square and output the board in Forsyth-Edwards Notation (F
```

```
27 fen = classify_cells(model, img_filename_list)
28
29 # Create and save the board image from the FEN
30 board = fen_to_image(fen)
31
32 # Display the board in ASCII
33 print(board)
34 # Display and save the chessboard image
35 board_image = cv2.imread('current_board.png')          cv2.imshow('current b
36
37 print('Completed!')
```

代碼鏈接：[https://github.com/andrewleeunderwood/project\\_MYM](https://github.com/andrewleeunderwood/project_MYM)

歡迎加入「從零開始學習SLAM」知識星球一起學習！

星球裡目前已經有接近**2200+人**。包含**三維重建**、**機器人SLAM**方向沉澱了2年多的干貨，包括大量**圖文**、**視頻教學視頻**、**實習/校招/社招機會**、**筆試面試題目及解析**、**疑難解惑**、**領域內最新開源論文/代碼/數據集**等（詳細介紹見後），還可以加入**粉絲專屬微信交流群/星球內部資料**、每天只需幾毛錢，即可快速觸達我們優質技術社區，一起交流學習，不管是剛剛入門的新手，還是已經是行業的從業者，都能獲益良多~

現在加入**立減30元**，優惠有效期至**12月29日晚上24點**截止，**名額有限，先到先得**，**3天內不滿意可全額退款**。



另外，我們推出的視頻課程：《**全網最詳細的ORB-SLAM2精講：原理推導+逐行代碼分析**》  
報名火爆，目前還有少量名額，現在加入知識星球可以享受**課程額外再立減100**的優惠！具體  
領取方式詳見星球內

# 全网最详细的ORB-SLAM2精讲

## 超实用的VSLAM解决方案

疑难点全面覆盖，带你深度掌握实际项目细节

原理：关键公式推导 + 走心示意图

代码：详细注释 + 逐行详解 + 源码勘误

主讲人：小六

中科院博士、多年VSLAM从业经验、计算机视觉life公众号创始人

長按或掃描二維碼查看課程介紹和購買方式：



專輯：計算機視覺方向簡介

專輯：視覺SLAM入門

專輯：最新SLAM/三維視覺論文/開源

專輯：三維視覺/SLAM公開課

專輯：深度相機原理及應用

專輯：手機雙攝頭技術解析與應用

專輯：相機標定

專輯：全景相機

## 交流群

歡迎加入公眾號讀者群一起和同行交流，目前有SLAM、三維視覺、傳感器、自動駕駛、計算攝影、檢測、分割、識別、醫學影像、GAN、算法競賽等微信群（以後會逐漸細分），請掃描下面微信號加群，備註：“暱稱+學校/公司+研究方向”，例如：“張三+上海交大+視覺SLAM”。**請按照格式備註，否則不予通過。**添加成功後會根據研究方向邀請進入相關微信群。請勿在群內發送廣告，否則會請出群，謝謝理解~

投稿、合作也歡迎聯繫：simiter@126.com



掃描關注視頻號，看最新技術落地及開源方案視頻秀 ↓

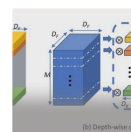


扫一扫二维码，关注我的视频号

喜歡此內容的人還喜歡

萬字綜述：用於深度神經網絡加速的Shift操作

極市平台



談一談我對AI項目落地的看法

極市平台



我國在校研究生的學術貢獻有多大？

3D視覺工坊

