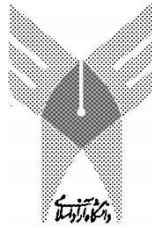


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه آزاد اسلامی
واحد تهران جنوب
دانشکده مهندسی برق
گروه مهندسی پزشکی بیوالکتریک

رشته: مهندسی پزشکی
گرایش: بیوالکتریک

عنوان:

تشخیص سلول های خونی با پردازش تصویر

نام استاد:

دکتر مهدی اسلامی

نام پژوهشگر:

زهرا حامدی طبری

برنامه تشخیص سلول های خونی با پردازش تصویر با زبان برنامه نویسی پایتون:

الزامات نرم افزاری

- Tensorflow-GPU نسخه های 2.2.0, 2.1.0, 2.2.0, 2.3.0 تست شده در نسخه های
- Pip install tensorflow-gpu (دستور نصب)
- TF-slim نسخه 1.1.0
- Pip install tf-slim==1.1.0 (دستور نصب)
- python 3.6 یا 3.7
- Cython نسخه 0.29.2
- Nvidia.Cuda.Toolkit
- Numpy
- Anaconda
- openCV
- visual studio c/c++

الزامات سخت افزاری

لیست gpu های سازگار با cuda جهت اجرای کتابخانه های پردازش تصویر

<https://developer.nvidia.com/cuda-gpus>

<https://developer.nvidia.com/cuda-legacy-gpus>

نحوه اجرای کد

برای شناسایی سلول های خونی، به سادگی فایل detect.py را در ترمینال یا با استفاده از

یک IDE اجرا کنید.

python detect.py

پوشه \data حاوی پنج تصویر آزمایشی است. و پوشه image شامل تصاویر پردازش شده خروجی است .

```
import cv2
import time
from utils import iou
from scipy import spatial
from darkflow.net.build import TFNet
```

در ابتدای برنامه ماژول های مورد نیاز را فراخوانی می کنیم .

```
def blood_cell_count(file_name):
    rbc = 0
    wbc = 0
    platelets = 0

    cell = []
    cls = []
    conf = []

    record = []
    tl_ = []
    br_ = []
    iou_ = []
    iou_value = 0

    tic = time.time()
    image = cv2.imread('data/' + file_name)
    output = tfnet.return_predict(image)
```

در بخش بعدی برنامه کلاس شمارش سلول های خونی رو فراخوانی و مقدار دهی میکنیم .
تابع time زمان اجرای برنامه را محاسبه و ذخیره میکند . مقادیر wbc,rbc,platelets به ترتیب برای نگه داری تعداد سلول های گلبول سفید و قرمز و پلاکت می باشد .
مقادیر iou برای حذف چند گانه اشیا از تصویر فراخوانی می شود .
تابع tic برای نگه داری زمان اجرای برنامه به طور کامل ، و تابع image از کتابخانه openCV مقدار دهی می شود و تابع output برای ذخیره مقادیر پیش تشخیص می باشد .

```
for prediction in output:
    label = prediction['label']
    confidence = prediction['confidence']
    tl = (prediction['topleft']['x'], prediction['topleft']['y'])
    br = (prediction['bottomright']['x'],
prediction['bottomright']['y'])

    if label == 'RBC' and confidence < .5:
        continue
    if label == 'WBC' and confidence < .25:
        continue
    if label == 'Platelets' and confidence < .25:
        continue
```

در این قست از برنامه اشیا (سلول های) شناسایی شده در توسط معماری yolo و مقادیرمختصات بالا سمت چپ پایین سمت راست برنامه جهت برچسب گذاری تصاویر فراخوانی و سپس بر اساس تخیص داده شده مقایسه می شوند و برچسب بر روی تصویر نهایی انجام میشود .

```
# clearing up overlapped same platelets
if label == 'Platelets':
    if record:
        tree = spatial.cKDTree(record)
        index = tree.query(tl)[1]
        iou_value = iou(tl + br, tl_[index] + br_[index])
        iou_.append(iou_value)

    if iou_value > 0.1:
        continue

    record.append(tl)
    tl_.append(tl)
    br_.append(br)

center_x = int((tl[0] + br[0]) / 2)
center_y = int((tl[1] + br[1]) / 2)
center = (center_x, center_y)
```

در این قسمت از برنامه به دلیل شکل متوالی پلاکت های خون احتمال همپوشانی وجود دارد که باعث تشخیص دو گانه و اشتباه می شود که این قسمت از برنامه در صورت همپوشانی نا صحیح مختصات همپوشانی را تشخیص داده ئ دوگانگی را حذف می کند .

```
if label == 'RBC':
    color = (255, 0, 0)
    rbc = rbc + 1
if label == 'WBC':
    color = (0, 255, 0)
    wbc = wbc + 1
if label == 'Platelets':
    color = (0, 0, 255)
    platelets = platelets + 1

radius = int((br[0] - tl[0]) / 2)
image = cv2.circle(image, center, radius, color, 2)
font = cv2.FONT_HERSHEY_COMPLEX
image = cv2.putText(image, label, (center_x - 15, center_y + 5),
font, .5, color, 1)
cell.append([tl[0], tl[1], br[0], br[1]])
```

این قسمت از برنامه با توجه به تشخیص نوع سلول های خون بر اساس برچسب گذاری مختصات تصویر که به صورت مربع تشخیص داده شده که شکل هندسی مربع اشغال فضای بالایی دارد و با فرمول تبدیل مختصات به دایره شکل دایره را با رنگ دهی براساس برچسب بر روی مختصات تشخیص داده شده در تصویر رسم دایره های قرمز و سبز و آبی به ترتیب برای گلبول قرمز و سفید و پلاکت انجام می شود .

```
if label == 'RBC':
    cls.append(0)
if label == 'WBC':
    cls.append(1)
if label == 'Platelets':
    cls.append(2)

conf.append(confidence)
```

در این قسمت بعد از اطمینان از پردازش تمامی موارد انجام شده مقدار دهی مورد نیاز را انجام می دهد .

```
toc = time.time()
pred_bb.append(cell)
pred_cls.append(cls)
```

```
pred_conf.append(conf)
avg_time = (toc - tic) * 1000
print('{0:.5}'.format(avg_time), 'ms')

cv2.imwrite('output/' + file_name, image)
cv2.imshow('Total RBC: ' + str(rbc) + ', WBC: ' + str(wbc) +
', Platelets: ' + str(platelets), image)
print('Press "ESC" to close . . .')
if cv2.waitKey(0) & 0xff == 27:
    cv2.destroyAllWindows()

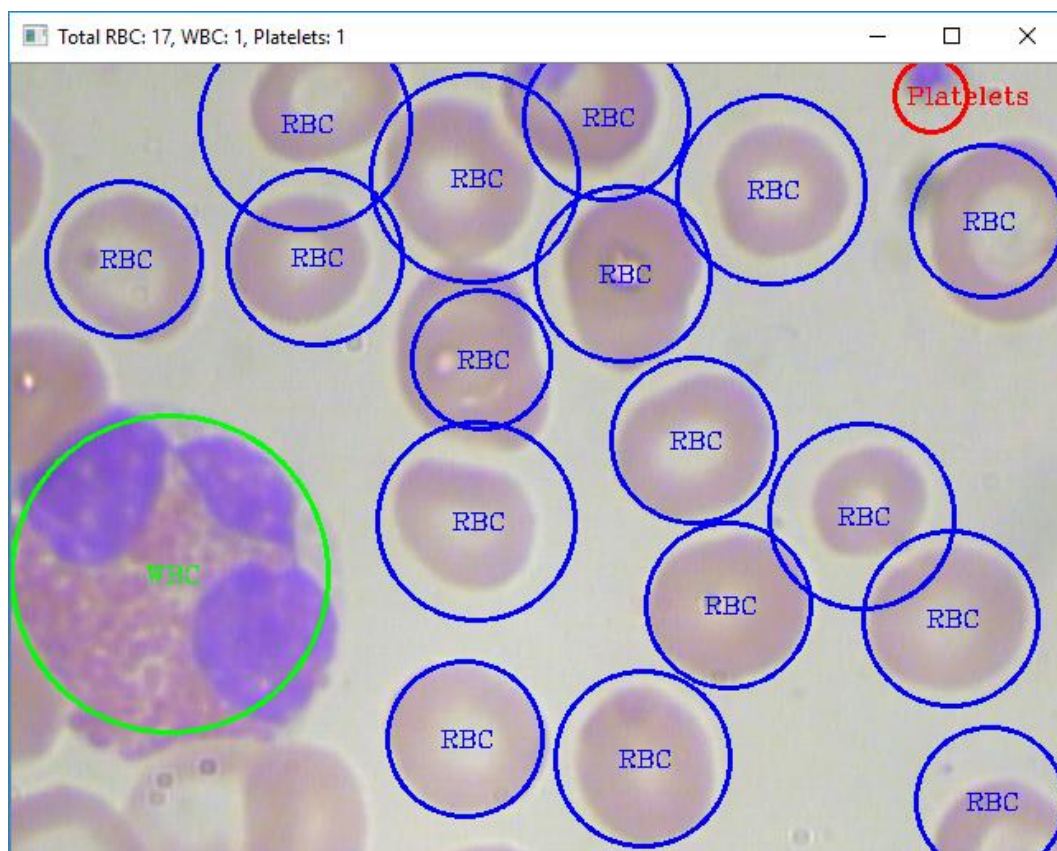
image_name = 'image_001.jpg'
blood_cell_count(image_name)

print('All Done!')
```

در این قسمت از کدهای برنامه مقدار زمان مورد نیاز برای پردازش برنامه محاسبه و تبدیل و سپس چاپ می شود . و بعد از آن فایل نهایی تصاویر پردازش شده ذخیره می شوند ، و در بخش بعدی از کاربر برای بسته شدن برنامه درخواست فشردن کلید ESC می شود .

و پس از نام گذاری تصاویر خروجی متن all done چاپ می شود .

نمونه تصویر خروجی تشخیص سلول های خونی



مراجع

Zhao J., Zhang M., Zhou Z., ET AL.: ‘Automatic detection and classification of leukocytes using convolutional neural networks’, Med. Biol. Eng. Comput., 2017, 55, (8), pp. 1287–1301

Habibzadeh M., Jannesari M., Rezaei Z., ET AL.: ‘Automatic white blood cell classification using pre-trained deep learning models: ResNet and inception’. AIP Conf. Proc., 2017, vol. 1883, pp. 1–8

Xu M., Papageorgiou D.P., Abidi S.Z., ET AL.: ‘A deep convolutional neural network for classification of red blood cells in sickle cell anemia’, PLoS Comput. Biol., 2017, 13, (10), pp. 1–27
He K., Zhang X., Ren S., ET AL.: ‘Deep residual learning for image recognition’. IEEE Conf. on Computer Vision and Pattern Recognition, December 2016

Cruz D., Jennifer C., Valiente L.C., ET AL.: ‘Determination of blood components (WBCs, RBCs, and platelets) count in microscopic images using image processing and analysis’. Proc. Int. Conf. on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), December 2017

Habibzadeh M., KrzyÅijak A., Fevens T.: ‘White blood cell differential counts using convolutional neural networks for low resolution images’. Proc. Int. Conf. on Artificial Intelligence and Soft Computing, 2013, pp. 263–274

Acharjee S., Chakrabartty S., Alam M.I., ET AL.: ‘A semiautomated approach using GUI for the detection of red blood cells’. Proc. Int. Conf. on Electrical, Electronics, and Optimization Techniques, 2016, pp. 525–529

Lou J., Zhou M., Li Q., ET AL.: ‘An automatic red blood cell counting method based on spectral images’. Proc. Int. Congress on Image and Signal Processing, BioMedical Engineering and Informatics, October 2016, pp. 1391–1396

Islam M.T., Aowal M.A., Minhaz A.T., ET AL.: ‘Abnormality detection and localization in chest X-rays using deep convolutional neural networks’, arXiv preprint arXiv:1705.09850v3, 2017

Avendi M.R., Kheradvar A., Jafarkhani H.: 'A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI', *Med. Image Anal.*, 2016, 30, pp. 108–119

Gulshan V., Peng L., Coram M., ET AL.: 'Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs', *J. Am. Med. Assoc.*, 2016, 316, (22), pp. 2402–2410

Ren S., He K., Girshick R., ET AL.: 'Faster R-CNN: towards real-time object detection with region proposal networks', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, 39, (6), pp. 1137–1149

Redmon J., Divvala S., Girshick R., ET AL.: 'You only look once: unified, real-time object detection'. *IEEE Conf. on Computer Vision and Pattern Recognition*, December 2016

Acharya V., Kumar P.: 'Identification and red blood cell automated counting from blood smear images using computer-aided system', *Med. Biol. Eng. Comput.*, 2018, 56, (3), pp. 483–489

Sarrafzadeh O., Dehnavi A.M., Rabbani H., ET AL.: 'Circlet based framework for red blood cells segmentation and counting'. *IEEE Workshop on Signal Processing Systems*, December 2015

Kaur P., Sharma V., Garg N.: 'Platelet count using image processing'. *Int. Conf. on Computing for Sustainable Global Development (INDIACom)*, March 20

