Using Machine Learning to Predict Changes of a Nuclear-Cytoplasmic Ratio to Assess Potential Malignancy

Joshua Wijaya, Minh-Khang Le PhD
Emerging Diagnostic and Investigative Technologies, Department of Pathology, Dartmouth Hitchcock Medical Center

Dartmouth Health

ABSTRACT

This research project explores the efficiency of using a machine-learning model to classify benign and malignant lung cells, which in turn is of use as high rates of lung cancer have been present as 1 in 16 males and 1 in 17 females seem to develop this cancer. Identifying malignant cells consists of taking samples of lung cells and creating whole slide images by scanning different areas of the retrieved cells. As malignancy of cells may be easily identified by researchers by looking at the nuclear-cytoplasmic ratios of cells, the sheer quantity of cells the researchers must work with is vast. Utilization of a machine-learning model can be used to diagnose all regions of malignancy from images provided to the model to decrease the amount of time necessary to identify regions of malignancy and potentially reduce the number of casualties by having information on exact locations of cells in a short duration.

INTRODUCTION

- About 10 million people die from cancer yearly, making it one of the largest concerns in the world
- Predictions allow clinicians to determine if patients should receive treatment
- If reliable predictions declare the patient clear of cancer cells, the patient may be discharged; this saves the hospital materials, reduces cost to the patient, and allows the clinician to have more materials/substances readily available for patients predicted to have malignancy

METHODS

Convolutional Neural Network

- Images of normal and malignant cells
- 15,000 images total; 12,000 for training and 3,000 for testing
  - Training and validation idea; 5 different folders for training images, and each folder would be used for training and validation

5-fold assignments

[97]:
```
root = fr"C:\Users\joshu\OneDrive\Pictures\EDIT_AI24\lung_colon_image_set\train_imgs"
n_img_eachfold = int(n_train/5)
indexes = range(len(train_imgs))
fold_indexes = random.sample(indexes, n_img_eachfold)
fold1  = [train_imgs[i] for i in fold_indexes]
remainings = [img for img in train_imgs if img not in fold1]

indexes = range(len(remainings))
fold_indexes = random.sample(indexes, n_img_eachfold)
fold2  = [remainings[i] for i in fold_indexes]
remainings = [img for img in remainings if img not in fold2]

indexes = range(len(remainings))
fold_indexes = random.sample(indexes, n_img_eachfold)
fold3  = [remainings[i] for i in fold_indexes]
remainings = [img for img in remainings if img not in fold3]

indexes = range(len(remainings))
fold_indexes = random.sample(indexes, n_img_eachfold)
fold4  = [remainings[i] for i in fold_indexes]
remainings = [img for img in remainings if img not in fold4]

indexes = range(len(remainings))
fold_indexes = random.sample(indexes, n_img_eachfold)
fold5  = [remainings[i] for i in fold_indexes]
remainings = [img for img in remainings if img not in fold5]
```

[111]:
```
for img in fold1:
    shutil.copy(img, os.path.join(train_root, "train1"))
```

[113]:
```
for img in fold2:
    shutil.copy(img, os.path.join(train_root, "train2"))

for img in fold3:
    shutil.copy(img, os.path.join(train_root, "train3"))

for img in fold4:
    shutil.copy(img, os.path.join(train_root, "train4"))

for img in fold5:
    shutil.copy(img, os.path.join(train_root, "train5"))
```

RESULTS

[57]:
```
# img_path = ['....\\train1', '....\\train2', ...]
class LungImage(Dataset):
    def __init__(
        self,
        root_dir,
        img_path: list(),
        preprocess=None,
                ):

        self.imgs = []
        self.labels = []
        self.root_dir = root_dir

        for path in img_path:
            path = fr"C:\Users\joshu\OneDrive\Pictures\EDIT_AI24\lung_colon_image_set"
            self.imgs += [os.path.join(path, img) for img in os.listdir(path)]

        #Task: figure out how to label self.imgs and put them inside self.labels

        self.label_mapping = {'norm': 0, 'aca': 1, 'scc': 2}
        root = fr"C:\Users\joshu\OneDrive\Pictures\EDIT_AI24\lung_colon_image_set\lung_image_sets"
        for file in os.listdir(root_dir):
            for label_name, label_idx in self.label_mapping.items():
                if label_name in file:
                    self.labels.append(label_idx)
        #End of task

        if preprocess is None:
            self.preprocess = transforms.Compose([ #preprocess defined
            transforms.ToTensor(),
            transforms.Normalize(mean=[0.485, 0.456, 0.406],std=[0.229, 0.224, 0.225])
            ])
        else:
            self.preprocess = preprocess

    def __len__(self):
        return len(self.imgs)

    def __getitem__(self, idx):
        img, label = self.imgs[idx], self.labels[idx]
        img = Image.open(img).convert("RGB")

        return img, label

from torchvision import transforms
preprocess = transforms.Compose([
            transforms.Resize((256, 256)),
            transforms.ToTensor(),
            transforms.Normalize(mean=[0.485, 0.456, 0.406],std=[0.229, 0.224, 0.225])
            ])
```

RESULTS

- Separate pathways to directories created
- Images transformed and resized to a 256x256 standard and RGB colored imaging
- Labels assigned to respective cells based on name of image in file

Acknowledgements: Minh-Khang Le PhD, Joshua Levy PhD