

# گزارش فاز دوم پروژه هوش محاسباتی

یوسف رضا زاده

علیرضا امینی

## بخش اول: محاسبه‌ی رتبه‌بندی نواحی استخراج شده از هر تصویر

ابتدا با استخراج ویژگی‌های هر تصویر و خوشه‌بندی این ویژگی‌ها، هر تصویر به 3 ناحیه تقسیم می‌شود. پس از استخراج ویژگی از هر ناحیه، بردار ویژگی هر ناحیه را تشکیل می‌دهیم. در این بخش در هر مرحله میانگین بردارهای هر تصویر را به دست آورده و به طبقه‌بندی می‌دهیم تا فرآیند یادگیری بر اساس آن‌ها صورت گیرد. سپس به ازای هر تصویر از میان نواحی باقی‌مانده، هر کدام را حذف کرده و در میانگین تاثیر نمی‌دهیم و با استفاده از تابع `predict_proba` احتمال تشخیص لیبل کلاس اصلی آن تصویر را محاسبه می‌کنیم و از تمامی آن‌ها ماکزیمم می‌گیریم تا ناحیه‌ای که کمترین تاثیر را در بالابردن احتمال لیبل کلاس آن تصویر را دارد شناسایی و حذف شود. این مراحل را تا زمانی تکرار می‌کنیم تا در هر مرحله کم اهمیت‌ترین ناحیه مشخص شود و به رتبه‌بندی نواحی برای هر تصویر برسیم. کد مراحل انجام شده در این حالت به صورت زیر می‌باشد:

```

def sort_region(images, super_pixels, features, labels, regions):
    priority, mean_features = [[] for i in range(super_pixels.shape[0])], []
    exist = [list(range(regions)) for i in range(super_pixels.shape[0])]
    clf = RandomForestClassifier(n_estimators=100, random_state=42)
    mean_features = [np.mean(f, axis=0) for f in features]
    clf.fit(mean_features, labels)

    for i in range(regions-1):
        for j in range(super_pixels.shape[0]):
            probs = []
            for k in exist[j]:
                current_exist = [v for v in exist[j]]
                current_exist.remove(k)

                f_mean = np.mean(features[j, current_exist, :], axis=0)
                f_mean_prob = clf.predict_proba(np.expand_dims(f_mean, 0))[0][labels[j]]
                probs.append(f_mean_prob)

            max_index = np.argmax(np.array(probs))
            priority[j].append(exist[j][max_index])
            exist[j].pop(max_index)
            mean_features[j] = np.mean(features[j, exist[j], :], axis=0)

    clf.fit(mean_features, labels)

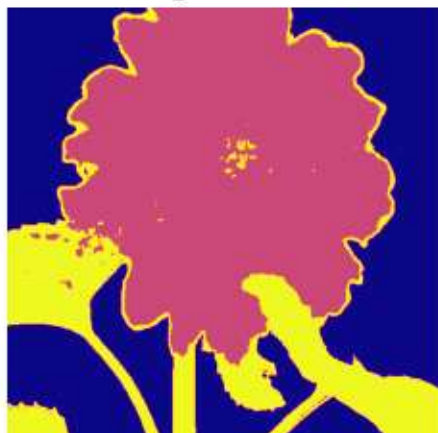
    for i in range(len(exist)):
        priority[i].append(exist[i][0])

    return priority

```

سپس مهم ترین ناحیه محاسبه شده برای هر تصویر را پلات می کنیم تا به صورت کیفی بررسی کنیم که آیا ناحیه بندی تصاویر به شکل مناسب انجام شده است یا خیر. چند مورد از خروجی ها به این صوت زیر می باشد:

Segmented



Most Important



Segmented



Most Important



Segmented



Most Important



## بخش دوم: انتخاب ویژگی

در این بخش ابتدا با استفاده از کل ویژگی‌های استخراج شده از مهم ترین ناحیه محاسبه شده در فاز قبل، یک طبقه بند بر روی داده‌ها فیت می‌کنیم. سپس با استفاده از تابع `predict_proba` بر روی طبقه‌بند آموزش دیده، احتمال لیبل کلاس هر تصویر را محاسبه و آن‌ها را `sort` می‌کنیم و 20 درصد از داده‌ها که پایین ترین احتمال برای آن‌ها به دست آمده بود را حذف می‌کنیم.

```
def select_data(datapoints, labels, rate):
    labels = np.array(labels)
    clf = RandomForestClassifier(n_estimators=100, random_state=42)
    clf.fit(datapoints, labels)
    sample_class_probs = []

    for p,l in zip(clf.predict_proba(datapoints), labels):
        sample_class_probs.append(p[l])

    sample_class_probs = np.array(sample_class_probs)
    select = int(sample_class_probs.shape[0] * rate)
    indexes = np.argsort(sample_class_probs)[:select]

    return datapoints[indexes], labels[indexes]
```

در بخش بعدی با 80 درصد باقی مانده داده‌ها ابتدا با هر ویژگی استخراج شده به تنهایی، طبقه‌بند را بر روی داده‌ها فیت می‌کنیم و دقت کل را به ازای هر ویژگی محاسبه می‌کنیم.

```
single_feature_accuracy = []

for i in range(selected_datapoints.shape[1]):
    mask, test_labels, test_preds, test_index, acc = classify(selected_datapoints[:, i].reshape(selected_datapoints.shape[0], 1), selected_labels, False)
    single_feature_accuracy.append(acc)

single_feature_accuracy = np.array(single_feature_accuracy)
print(single_feature_accuracy)
```

سپس ویژگی با بالاترین دقت را به عنوان بهترین ویژگی انتخاب می‌کنیم و در هر مرحله برای ویژگی‌های بعدی میزان عدم شباهت با ویژگی‌های انتخاب شده را محاسبه می‌کنیم و با استفاده از این پارامتر و همچنین میزان دقت محاسبه شده برای آن ویژگی به تنهایی، معیار `f1-score` را محاسبه و به عنوان امتیاز هر ویژگی قرار می‌دهیم. ویژگی با بالاترین امتیاز به عنوان بهترین ویژگی در هر مرحله انتخاب می‌شود.

```

best_index = np.flip(np.argsort(single_feature_accuracy))
selected = []
last_acc = 0
not_better_step = 0

while 1:
    if len(selected) == 0:
        selected.append(best_index[0])
    else:
        corrs = []

        for i in range(best_index.shape[0]):
            feature_corrs = []
            for j in range(len(selected)):
                feature_corrs.append(np.abs(np.corrcoef(selected_datapoints[:, i], selected_datapoints[:, selected[j]])) * 0.1))
            corrs.append(1. - np.array(feature_corrs).max())

        corrs = np.array(corrs)
        feature_scores = (1 * corrs * single_feature_accuracy) / (corrs + single_feature_accuracy)
        best_scores = np.flip(np.argsort(feature_scores))

        for idx in best_scores:
            if not idx in selected:
                selected.append(idx)
                break

    mask, test_labels, test_preds, test_index, acc = classify(selected_datapoints[:, selected].reshape(selected_datapoints.shape[0], len(selected)), selected_labels, False)

    if acc - last_acc < 0.01:
        not_better_step += 1
    else:
        not_better_step = 0

    if len(selected) == selected_datapoints.shape[1] or last_acc - acc > 0.02 or not_better_step == 2:
        break
    else:
        last_acc = acc

selected = selected[:-1]

```

در هر مرحله انتخاب ویژگی مجددا طبقه‌بند را با ویژگی‌های انتخاب‌شده تا آن مرحله بر روی داده‌ها فیت می‌کنیم و دقت کل را محاسبه می‌کنیم. شرط توقف در هر مرحله نیز به این صورت خواهد بود که اگر دقت محاسبه شده در هر مرحله از دقت محاسبه‌شده در مرحله قبل به اندازه یک مقدار آستانه (مقدار استفاده‌شده = 0.02) کمتر بود و یا در چند مرحله متوالی تغییرات دقت جدید از دقت قبلی از یک مقدار آستانه دیگر (مقدار استفاده‌شده = 0.01) کمتر بود، انتخاب ویژگی متوقف می‌شود و یا اینکه همه ویژگی‌های ممکن توسط الگوریتم انتخاب شوند.

در نهایت این الگوریتم با مقادیر آستانه‌های ذکرشده و برای 2 مرحله عدم بهبودی اجرا شد و ویژگی‌های میانگین و انحراف معیار کانال‌های h و s و v و همچنین انحراف معیار X پیکسل‌ها به عنوان بهترین ویژگی‌ها انتخاب شدند.