

Relatório MAC0318 EP3

Renato Lui Geh — NUSP: 8536030

1 Classificador

Para a classificação, foi usada uma “amostra” das imagens. Para treinar pelas regras, usou-se as médias das intensidades dos pixels de certas regiões selecionadas. Chame de R uma região da imagem. Foi atribuída uma média $\mu_R(D)$ onde D é a amostra das imagens, separando as imagens pelos seus respectivos rótulos de classificação.

Para classificar uma imagem I , computou-se as médias das mesmas regiões que na amostragem. Seja $\mu_{R_1}, \mu_{R_2}, \dots, \mu_{R_n}$ as médias computadas nas amostragens e $\hat{\mu}_{R_1}, \hat{\mu}_{R_2}, \dots, \hat{\mu}_{R_n}$ as médias computadas na imagem I para classificação. O erro acumulado E_I é dado por:

$$\sum_{i=1}^n |\mu_{R_i} - \hat{\mu}_{R_i}|$$

Por fim, queremos achar:

$$\arg \max_{l \in L} E_I = \arg \max_{l \in L} \sum_{i=1}^n |\mu_{R_i} - \hat{\mu}_{R_i}|$$

Onde L é o conjunto de classes.

2 Estratégias

Para achar as regiões R mencionadas na seção anterior, foram escolhidas as seguintes regiões da imagem:

1. Matriz triangular L na i -ésima diagonal;
2. Matriz triangular U na i -ésima diagonal;

3. “Reflexo” da matriz triangular L na i -ésima diagonal;
4. “Reflexo” da matriz triangular U na i -ésima diagonal;
5. Regiões retangulares de dimensão (s_x, s_y) ;
6. Partições retangulares principais;
7. Quadrantes;
8. “Teto” e “chão”.

Para as matrizes triangulares, usou-se `np.tril` e `np.triu`. Sejam L_i e U_i as matrizes triangulares L e U respectivamente na diagonal i . O “reflexo” da matriz triangular L_i e U_i são dadas por $M - L_i$ e $M - U_i$ respectivamente, onde M é a matriz da imagem original.

As regiões retangulares são todas as regiões de tamanho (s_x, s_y) nas posições $(i \cdot s_x, j \cdot s_y)$, com $i \in \mathbb{Z}_w, j \in \mathbb{Z}_h$, onde (w, h) são as dimensões da imagem.

As partições retangulares principais são as regiões:

1. $(0, 0, w/2, h/2)$
2. $(0, 0, w/2, h)$
3. $(0, h/2, w/2, h)$
4. $(w/2, 0, w, h/2)$
5. $(w/2, 0, w, h)$
6. $(w/2, h/2, w, h)$

Os quadrantes são dados pelas regiões abaixo, com $i \in \{2, 3, 4, 5\}$:

1. $(0, 0, w/i, h/i)$
2. $(0, h - h/i, w/i, h)$
3. $(w - w/i, h, w, h - h/i)$
4. $(w - w/i, 0, w, h/i)$

Para o “teto” e “chão”, escolheram-se as regiões “mais pra baixo” e “mais pra cima” da imagem. Seja $3 \leq i \leq 10$, com $i \in \mathbb{N}$.

1. $(0, h - h/i, w, h)$
2. $(0, 0, w, h/i)$
3. $(0, h/i, w, h - h/i)$
4. $(w - w/i, 0, w/i, h)$
5. $(0, 0, w/i, h)$
6. $(w/i, h/i, 0, h)$
7. $(w/i, h/i, w - w/i, h - h/i)$

3 Resultados

A acurácia não foi muito boa. Ficou em $\approx 68\%$.

4 Outras estratégias

Pensei também em duas outras coisas que poderiam melhorar: separar as médias em quantis e dar pesos às regiões.

Para a separação em quantis, a ideia era de, ao invés de tomar a média de todas as intensidades, separar as intensidades em quantis, para que valores extremos fossem tomados em consideração, e não fossem absorvidos pela média. Para isso, ao invés de computar a média diretamente, as médias foram separadas em clusters. Rodou-se k-means com q clusters, e para cada cluster q_i , computou-se a média. Na classificação, tomava-se em consideração o erro:

$$E_I = \sum_{j=1}^q |q_j| \left(\sum_{i=1}^n |\mu_{R_i}(q_j) - \hat{\mu}_{R_i}(q_j)| \right)$$

Onde $\mu_{R_i}(q_j)$ é a média da região R_i apenas no cluster q_j . Nos resultados, usando esta estratégia, aumentou-se em apenas 1 a 2% de acurácia, mas demorou muito mais para treinar e classificar.

A outra estratégia era dar pesos às regiões. A partir de um vetor $W = (w_1, \dots, w_n)$, foi criada uma matriz diagonal \mathbf{W} em que $\mathbf{w}_{ii} = w_i$ e todas as outras entradas são zero. O erro então é dado por:

$$E_I = \sum (M - \hat{M})^T \cdot \mathbf{W}$$

Onde M e \hat{M} são as médias da amostra e da imagem I em forma matricial respectivamente. No final, achar bons valores para W foi o problema. Além disso, a multiplicação de matrizes acima demorava muito tempo no Raspberry Pi.