Project Pipeline for Image Classification

PROJECT:

https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/iet-ipr.2015.0385

OVERALL PIPELINE:

1) Preprocessing, Segmentation

- Experiment with a few images first, adapt.
- Use a for loop with the selected methods on all images.
- Save the processed images.

2) Extract Features

- Load processed images.
- Look at feature distributions, adapt.
- Use a for loop for feature extraction.
- Save the extracted features.

3) Classification + Evaluation

- Load the extracted features.
- Recommended classifiers for already-extracted features: nearest mean, nearest neighbor, decision tree, logistic regression, random forest.
- For classifiers relying on distances, scale the features using StandardScaler from scikit-learn.
- Train/test classifiers on different parts of the data.
- Aim for similar performance on training and validation sets.
- If the classifier overfits (does not generalize to unseen data), it could be due to too few samples, a too complex classifier, or too many features.
- Analyze classifiers, adapt features and/or classifiers.
- Split data into K parts (Stratified cross-validation: keep class proportions).
- Select the best method based on average validation performance.
- Test on held-out data (only for reporting).

4) Final Evaluation

- Train/test "final" classifiers (no adjusting of the method after this).
- Visualize the trained classifier (see `https://stackoverflow.com/questions/41138706/recreating-decision-bound`
- Evaluate accuracy: true labels vs probabilities (0.5 is the default threshold, but it does not have to be).
- Evaluate the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC). AUC of 1 is perfect, 0.5 is a random guess.
- Save the probabilities (outputs of the classifiers).
- Save the classifiers.
- Load the probabilities and classifiers.
- Repeat the experiment several times, shuffling the data.
- Report mean and standard deviation.
- With probabilities, calculate other evaluation metrics.
- Evaluate incorrectly classified examples (look at the "most incorrect" ones). Are there any patterns?
- Evaluate training and validation performance at multiple training set sizes (learning curves).
- With the classifiers, classify other datasets.

To Be Implemented: Hair Detection

- Assess how much hair there is (use manual scores to check that the method works correctly).
- Create an extended classification method that uses hair extraction features.

BASELINE FEATURES:

- Good features should have variations in values for different images and not be highly correlated with other features.
- For analysis, use scatterplots and distributions (should be normal). The best features will have the least overlap.
- GitHub repo for ABC features: `https://github.com/raumannsr/ENHANCE`.
- **Asymmetry:** Fold the shape and look at non-overlapping parts.
  `def asymmetry(mask)`.
- **Border:** Define border feature.
- **Color:** Use simple Linear Iterative Clustering (LIC) segmentation algorithm.
  `def get_multicolor_rate(im, mask, n)`.

EXTENDED FEATURES:

- **Compactness:** $c = \frac{\text{perimeter}^2}{4\pi \cdot \text{area}}$.
- To calculate perimeter, create a mask that is slightly smaller (use morphology). Subtract the smaller mask from the larger one.
  `def get_compactness(mask)`.
- **Blue-White Veil:** Structureless blue blotches with an overlying whitish haze.
  `def measure_blue_veil(image)`.
- **Irregular Dots/Globules:**
  `def measure_globules(image)`.
- **Atypical Pigment Network:** Reticular lines, heterogeneous for color and thickness.
  `def measure_irregular_pigmentation(image)`.

- **Atypical Vascular Pattern:** Linear, dotted, or globular vessels irregularly distributed. `def measure_vascular(image)`.
- **Regression Structures:** White scar-like depigmentation or peppering. `def measure_regression(image)`.

`https://dermoscopedia.org/Seven\_Point\_Checklist`

Steps for Image Processing:

- **Peak Signal to Noise Ratio (PSNR):** Measures the average mismatch between pixels. Acceptable image quality is achieved if SSIM ¿ 0.8.
- **Structural Similarity Index (SSIM):** Better aligns with human perception. Acceptable image quality is achieved if PSNR ¿ 20 decibels.

Processing Pipeline:

- Read image (batch size, shuffle?).
- Convert RGB to grayscale.
- Image denoising.
- Histogram equalization.
- Segmentation mask for the lesion (some provided).
- Convert grayscale to binary (bimodal distribution / Otsu's method).
- Apply erosion, then dilation (closing) to remove isolated pixels.
- Apply dilation, then erosion (opening) to close small holes.

Hair Detection:

- Common edge detection techniques: Sobel, Canny, LoG.
- Top-hat transform: $I_{\text{original}} - I_{\text{opening}}$.
- Black-hat transform: $I_{\text{closing}} - I_{\text{original}}$.
- Extract darker "breaks" over a bright area (skin).

Region Growing:

- Binarizing by taking connectivity into account.

Evaluate Quality of Annotations:

- Accurate and consistent.
- Accurate but not consistent.
- Not accurate but consistent.
- Not accurate and not consistent.

Metrics:

- **Accuracy:** Are we measuring the thing we wanted?
- **Consistency:** Do we always measure the same thing?

Agreement Metrics:

- **Cohen's Kappa:** Analyzes pairwise agreement.
- **Dice Coefficient:** A = ground truth (e.g., expert), B = segmentation algorithm. Range: 0 (no overlap) to 1 (perfect match).