Animation character identification from color images

Alexis Vallet, Yuki Nakagawa, Hiroyasu Sakamoto

Kyushu University, University of Technology of Belfort-Montbéliard

October 2, 2013

- (Semi) supervised classification of animation character images.
- Dealing with variations in character posture, occlusion, drawing style, exaggerations.
- Application domain: web artist communities such as Pixiv, deviantArt.

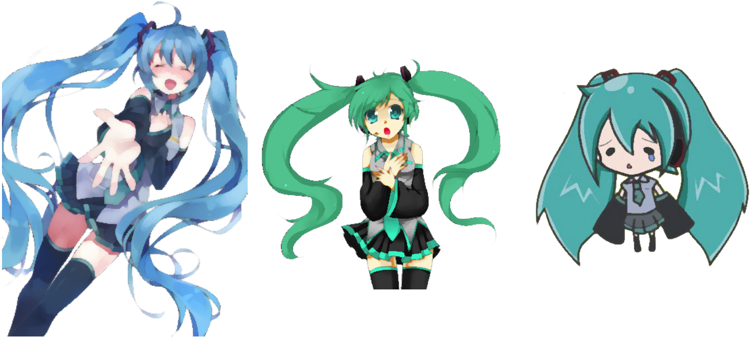


Figure : Images illustrating variations for a single character.

- Preprocessing: removing outlines, switching color space.
- Segmentation to isolate parts of interest - hair, clothes, face...
- Classification by comparing segmentation against training set.

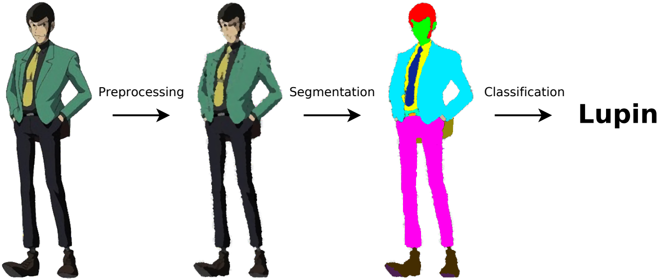
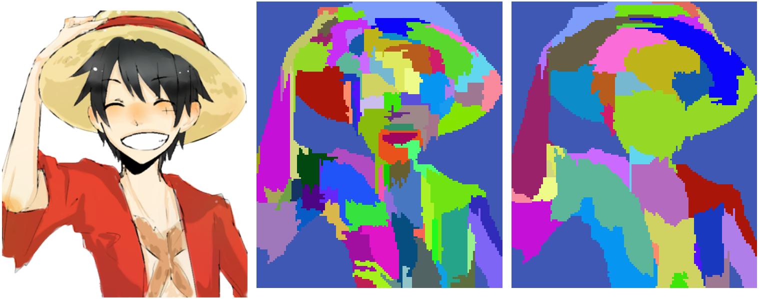


Figure : Diagram depicting how preprocessing, segmentation and classification interact.

Felzenszwalb' segmentation [**felzenszwalb2004efficient**]

- Graph method based on Kruskal's algorithm.
- Efficient: $O(n\log(n))$ time with 4-connected neighborhood.
- Accurate: neither too "coarse" nor too "fine".
- But depends on a scale parameter k which controls the size of segments.

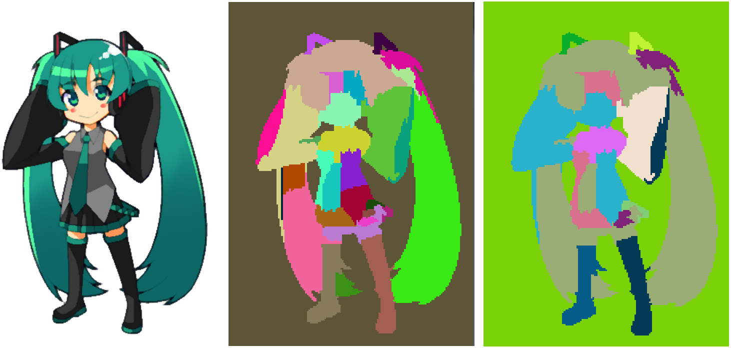


(a) Original image

(b) $k = 100$.

(c) $k = 1000$.

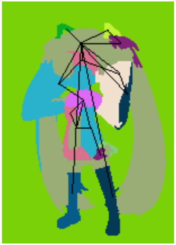
- Post processing by merging segments with close hue.
- Allows varying segment sizes and non connected segments.



(a) Original image. (b) Before merging. (c) After merging.

Spectral classification method

- For segmentation S consider features $(f_i : S \to \mathbb{R}_i^q)_{1 \leq i \leq m}$. (average color, gravity center, size...)
- For each feature f_i, compute K-nearest neighbor graph G_i on S with weights $w(u,v) = e^{-\frac{||f_i(S_u) - f_i(S_v)||^2}{\sigma_i^2}}$ and Laplacian L_i.



(a) Example of graph on S.

$$L_i(u,v) = \begin{cases} \sum_{u' \text{ adjacent to } u} w(u,u') & \text{if } u = v \\ -w(u,v) & \text{if } u \text{ and } v \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

(b) Laplacian matrix definition.

- Only use the eigenvectors from the k smallest nonzero eigenvalues of L_i.
- Use method from Wilson, Hancock, Luo to create pattern vectors B_i from these eigenvectors [**wilson2005pattern**].
- Concatenate into feature vector $B = (B_1^T \dots B_m^T)$, classify using SVM.

Results and analysis

- Low recognition rate (close to random).
- Graphs do not encode enough information about individual segments.
- Deals poorly with different number of segments.

Segment matching classification

- Consider 3 features for each segment: average $L^*a^*b^*$ color, gravity center, and area.
- Measure similarity between segments using a fuzzy system.
- Find a one to one relation between similar segments of 2 images.

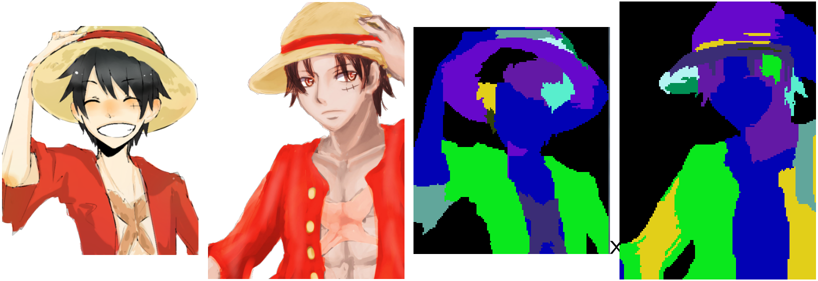
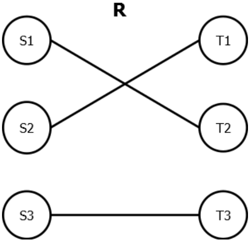


Figure : Original images (left) and corresponding relation (right). Segments with the same color are matched together.

- Measure overall similarity $sim(S,T)$ between segmentation S and T by sum of matching segments similarity weighted by segment areas.
- Classify by nearest neighbor.

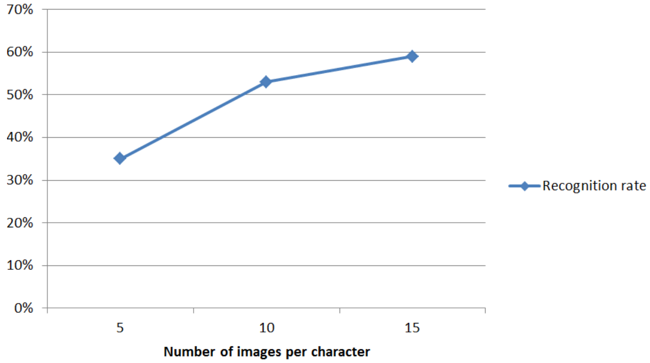


$$sim(S,T) = \sum_{(S_i,T_j)\in R} (|S_i|+|T_j|)s_{ij}$$

Where s_{ij} denotes the similarity between segments S_i and T_j by the fuzzy system.

Results and analysis

- 59% recognition rate for dataset with 12 characters and 15 images per characters.
- Recognition rate scales well with size of dataset.
- Has trouble with characters sharing similar color palette.



Possible extensions:

- Color palette issues: determining a (possibly non-linear, or high-dimensional) color space ideally separating training data, with some (semi) supervised embedding method [**urahama2007semi**] ?
- Background extraction: detecting important character features (face, hair, clothes) using method inspired by the face detection algorithm from Viola and Jones [**viola2004robust**] ?
- Also using segmentation graph, as in works from Bach and Harchaoui [**harchaoui2007image**] ?

References