

Learning Image Anchor Templates for Document Classification and Data Extraction

Prateek Sarkar

Perceptual Document Analysis Area

Palo Alto Research Center, Palo Alto, California, USA

psarkar@parc.com

Abstract

Image anchor templates are used in document image analysis for document classification, data localization, and other tasks. Current tools allow human operators to mark out small sub-images from documents to act as anchor templates. However, this requires time, and expertise because operators have to make informed decisions based on behavior of the template matching algorithms, and the expected degradations patterns in documents. We propose learning templates for a task automatically and quickly from a few training examples. Document classification or data localization can be done more robustly by combining evidence from many more discriminating templates (e.g., hundreds) than would be practicable for operators to specify.

1 Introduction

Template matching techniques for images have been widely used for object recognition and localization. Tauschek’s “Reading Machine” applied template matching for character recognition [13]. Eight decades later a textbook on template matching in computer vision was just published [2]. In document image analysis, template matching has been used heavily in optical character recognition, layout style matching, image registration (e.g., by locating fiduciary marks), document type classification, logo detection, and locating anchor points for data extraction. Image templates find favor in many industrial document image analysis solutions because they are easily visualized, and fast matching algorithms are available.

We focus on two applications of template matching: doctype classification and field localization for data extraction. In doctype classification, each document category is characterized by one or more templates that match images of that category. Detection of the target category is then accomplished by voting or some combination of the results of template matching on new images. In data extraction from forms, data fields are located by first finding templates describing the fixed

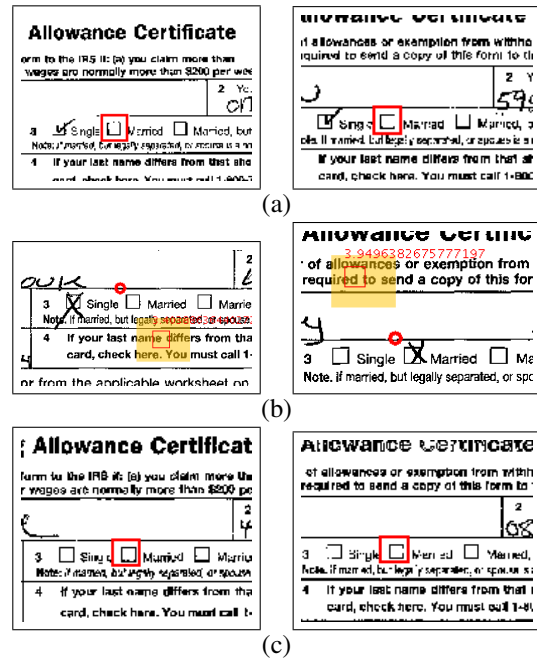


Figure 1. Data extraction with learned IATs. (a) An operator marks a field of interest in a few example images by dragging rectangles. (b) IATs are learned automatically to localize the upper left corner of the desired field. (c) A composite of 40 learned IATs are applied to locate the field in new forms.

form content. Such *image anchor templates* (IAT) can also be used for form registration and skew correction. Typically IATs are designed by human operators by carving out small image regions that can be used as templates. Robustness is achieved by relying on multiple templates. However, picking the best templates for a given job is both labor and skill intensive. Operators must learn, over time, the nuances of document degradation types and behavior of template matching functions. Still the most salient parts of images are not necessarily the most discriminative or informative for the task at hand - so operators have to revise their intuition based on empirical tests.

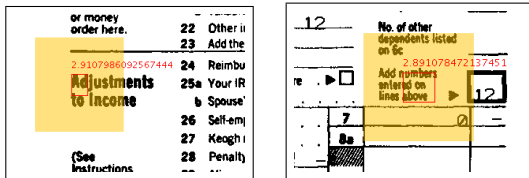


Figure 2. Two learned templates to detect Form 1040 (page 1) automatically learned from a few (3-5 examples per category) tax forms in the NIST data set. The template sub-image is shown with a red outline, the search region is shown in translucent orange, and supervised template quality score in red font.

To alleviate the undue burden on operators, we propose automatic learning of templates for a given task from image data. Figures 1,2 illustrate the results of our approach on the two tasks. All shown examples are from the publicly available NIST data set. Visualizing templates ranked by quality scores shows us that seemingly similar templates can be very different scores countering initial intuitions.

2 Template matching algorithms

Many different template matching schemes can be found in image processing textbooks. Most well known are correlation techniques, and speed-ups. Simple correlations are very sensitive to variations due to sampling and geometric transformations [9]. Many pixel-weighting schemes are also known to account for degradation patterns, correlations, and discriminative power [6, 8]. An interesting departure from correlation based templates is the idea of n-tuples [5] as transformation-robust signatures of rigid patterns. Another form of fast pattern matching for localizing words was proposed by Spitz [12]. Locating repeated instances of local appearance by a clever hashing technique specifically designed for document images was described in [10]. Deformable templates address non-rigid transformations of images [1, 14], but are more popular in computer vision applications than in document image analysis. A learnable deformable template model based on locational distributions of low level image features (such as strokes, edges) was applied to classify entire page images in [11]. A general area of research is in matching images not on the basis of rigid shape but by other aspects of appearance e.g., texture. Local histogram based matching techniques are likely to be very useful here [7].

For our work, we chose to use the efficient Hausdorff distance based template matching scheme of Rucklidge [4]. But the reader should note that our algorithms should work unchanged for any form of template matching such as the ones mentioned above.

3 A generic definition and notation

In order to facilitate further discussion we start with a few notations and definitions. Traditionally an *image anchor template* is a small image, so that we can look for matches in a target image I under a set of *valid transformations*. Typically for efficiency only translations are allowed, but sophisticated matching algorithms may allow rotations, scaling or affine transformations. We generalize this concept to define an image anchor template as a function $A(I) = (m, t)$ where t is a valid transformation, and $m \in [0, 1]$ is the *match-score* of the template to the image I transformed by t . Higher scores indicate better matches. For convenience, we shall treat m and t as functions themselves. Typically the match score is compared to some threshold q to produce a boolean template detection function $d(I; q) = 1[m(I) > q]$.

This definition is general enough to incorporate a wide variety of anchor templates including image based, and text-content based templates. In our experiments an image template is represented by a small image (or sub-image of a larger image), the valid transformations are restricted to translations within a region of interest (ROI), and the matching function is provided by the efficient Hausdorff Matching algorithm [4].

Composite templates: Multiple image anchor templates A_i may be combined to form a composite template. Some document analysis systems provide simple voting, counting, or Boolean combinations of multiple templates as a feature. We have explored two other kinds of composition. For document classification, we use the match scores m_i from a collection of templates as features, and use an Adaboost [3] classifier output as the composite match-score. For data-extraction, we use a simple weighted average of location predictions t_i as the composite location prediction. Combining hundreds of automatically learned templates in this way provides robustness against image variations both accidental (degradations) and by design (multiple variants of same form.)

4 Learning image anchor templates

4.1 Template Scoring

For any given task, some templates are better than others. Any image together with a matching function can be used to construct an IAT. But not all such IATs will be good at distinguishing same-class document images from other documents. Further, given any template image, different matching (or detection) functions will return different results on the same set of target images.

When documents are rigidly formatted (such as forms), salient sub-images carved out of one image will match other similar images. Current tools allow human operators to specify sub-regions of images to use as indicator templates for a category. However, picking the

best templates is not an easy task – experienced operators understand quirks of template matching functions, as well as those of document degradation patterns. Even more difficult is to find templates that are not likely to fire on exemplars of a different category. This is especially true if documents to be considered have fixed and variable parts, and some parts may be shared by multiple categories. For example, the same company logo may be a very salient feature on different forms, letter heads, and invoices. Since the discriminative power of an IAT is essentially one to be estimated empirically, it would be useful to have an automated way to rank templates based on their discriminative power.

Supervised discriminative quality: Let *positive exemplars* be a collection of document images that we expect an IAT to match, and *negative exemplars* be a collection that ought not be matched by a template. For any anchor template A , we can compute m for all the exemplars, and the ROC curve by varying the acceptance threshold q in $d(I; q)$. We then compute the following quality-score for the template and matching function s as:

$$s(A; \alpha) = \text{best}_{ROC}(A; \alpha) + \text{area}_{ROC}(A) + \text{margin}(A)$$

Here $\text{area}_{ROC}(\cdot)$ is the area under the ROC curve, and for a *relative false alarm penalty* of α , $\text{best}_{ROC}(A; \alpha) = \max_q(\text{hit}(q) - \alpha \cdot \text{falseAlarm}(q))$.

If a template detection function $d(A; q)$ perfectly discriminates between positive and negative exemplars then both $\text{best}_{ROC}(\cdot)$ and $\text{area}_{ROC}(\cdot)$ are 1. Among all such perfectly discriminating templates, we would prefer (assign higher quality to) those with higher margins. The margin is defined as the difference between the smallest matching score for a positive exemplar and the largest matching score for a negative exemplar. Following the definition of matching scores, the largest possible margin is 1.0. Hence our empirical supervised template quality score is zero at worst and 3.0 at best.

Empirical one-sided repeatability quality: In the above, positive and negative classes are symmetric with respect to the template matches, i.e., templates simply needed to be discriminative. But we assert that templates must match positive exemplars and not match negative exemplars. This enables us to discard many candidate templates after testing them only on positive exemplars, thus speeding up the search.

We measure this *one-sided* match quality by

$$o(A = (m, t)) = \sum_{I \in \omega_P} p(\omega_P | m(I))$$

where ω_P and ω_N denote the positive and negative categories respectively. Each probability on the right-hand side is computed using Bayes' rule with equal priors starting from the following formulae: $p(m(I) | \omega_P) \propto e^{-\lambda_P \cdot m(I)}$, and $p(m(I) | \omega_N) \propto e^{-\lambda_N (1-m(I))}$. The parameters λ can be trained, but for space considerations we

restrict ourselves to $\lambda_P = 1$, $\lambda_N = 10$ giving the negative category a flatter distribution.

Empirical one-sided anchoring quality: For anchoring (localizing) data-fields in a data extraction task, we would like a good template to be *repeatable* (high match-scores when field is present), *discriminative* (low match-scores when field is absent), and highly predictive of the field location (*anchoring power*). Ignoring discriminability for now, we combine repeatability, and anchoring power into a single quality score:

$$oa(A) = \sum_k p(\omega_P | m(I_k)) * e^{\sum_{j <= k} (x_k - u_j)^2 / k}$$

where I_k are images in the positive category, with the index k such that the images are sorted in decreasing order of $m(I_k)$.

4.2 Exploring candidate templates

With the above setup in place, our template learning algorithm for a given task comprises of generating many template candidates, ranking them according to the appropriate metric, and retaining the top few (for example the top 200 templates per category for document classification.) These templates can then be used to form composite templates for the classification or data extraction task.

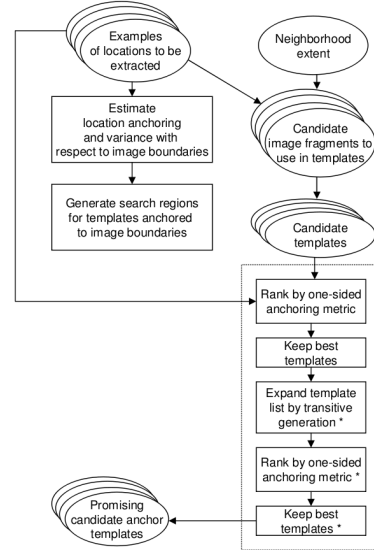


Figure 3. Schematic of the algorithm for finding the best anchoring templates. Sections marked by asterisk (*) indicate optional stages that be left out or expanded even further to explore if slight alterations would yield better templates.

Let us take, for example, the classification of structured form images. Any small sub-image from a positive exemplar can be used to form an IAT. However, for even modest sized training sets, the number of such sub-images is prohibitive. The challenge is to limit the search space effectively. We do this by controlled iterative expansion and refinement of a list of candidate

templates. We first limit our search to templates formed from sub-images of a single randomly picked positive exemplar (*grid-based sub-image generator*.) These candidates are then applied only to a few positive exemplars, and ranked according to the one-sided repeatability metric. Only the top templates are used as seeds to obtain new candidates by perturbation of the template, and *transitive exploration*. In the latter, for each candidate template, we introduce as competing candidates sub-images carved from around good match-locations in other images. The idea is that these candidates are all similar, but slightly different (for example, one may have less noise or extraneous marking than another) and potentially better than our initial candidate. Another way to expand candidate templates is to start the process anew, with positive exemplars that are not matched by the latest short-list of template candidates. This helps us cover multiple variants of documents in the positive category. Clearly some of these techniques are unique to image templates, others are applicable to the generic template definition. The strategy for learning templates for data anchoring is similar (Figure 3.)

5 Experimental validation

We have implemented the anchor template learning system in Java, and this has been tested on a couple of document categorization tasks. For fixed format document categories, such as in the case of NIST forms, the system can learn discriminating templates from as few as two or three exemplars per form category. Human identification of IATs is time consuming, but automatic template learning allows us to identify, and apply hundreds of templates per category for both document classification and data anchoring. Combining evidence from a large number of templates results in robust performance, even with little training data. Trained on 3-5 exemplars per form category, we get perfect classification of the entire NIST data set. On a private data set (also thousands of documents) with more unstructured image categories (such as hand-written notes) we notice that IAT based document classification provides 100% precision at >95% recall rates for fixed-form categories, while other categories are reliably rejected. *While this is expected from IAT based systems, the biggest gain is that automatic training takes only 20 seconds to a few minutes per category, whereas experienced operators report taking up to a few days to setup for the same classification task because of necessary trial and error runs.*

Similarly in our prototype data extraction application on NIST forms, it is sufficient for the operator to identify five to ten exemplars of a field of interest, before the system learns and continues the task automatically and robustly (again based on tens of automatically learned templates) on hundreds of forms of the same kind. In our setup, each matching template predicts

the location of the field, and the average of these predictions weighted by the match-score is the composite prediction. We declare failure on a document if there is high variance among the individual location predictions. Since we combine evidence from many templates, the system is quite robust, and the few failures we observed among hundreds of forms were fixed when we increased the range of valid transformations for each template. This increased learning and data localization times, but still several pages can be processed per second.

References

- [1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, October 1997.
- [2] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. John Wiley and Sons, 2009.
- [3] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journ. Computer and System Sciences*, 55(11):119–139, 1997.
- [4] D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff distance. Technical Report TR 92-1321, Cornell University, 1992.
- [5] D. Jung, M. Krishnamoorthy, G. Nagy, and A. Shapira. N-tuple features for OCR revisited. *IEEE Trans. PAMI*, 18(7):734–745, July 1996.
- [6] G. Kopec. Multilevel character templates for document image decoding. In L. Vincent and J. Hull, editors, *Document Recognition IV, Proceedings of SPIE, vol. 3027*, 1997.
- [7] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Trans. on PAMI*, 31:2129–2142, 2009.
- [8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, November 1998.
- [9] G. Nagy. Classification algorithms in pattern recognition. *IEEE Trans. Audio and Electroacoustics*, 16(2):203–212, June 1968.
- [10] T. Nakai, K. Kise, and M. Iwamura. Hashing with local combinations of feature points and its application to camera-based document image retrieval. In *Proc. CB-DAR05*, page 8794, 2005.
- [11] P. Sarkar. Image classification: Classifying distributions of visual features. In *Proc. of the 18th ICPR*, Hong Kong, 2006.
- [12] A. L. Spitz. Shape-based word recognition. *Intl. Journ. Document Analysis and Recognition*, 1(4):178–190, May 1999.
- [13] G. Tauschek. Reading machine. U. S. Patent 2026329, Dec 1935. Appl. May 1929.
- [14] A. L. Yuille, D. S. Cohen, and P. W. Hallinan. Feature extraction from faces using deformable templates. In *Proc. CVPR*, pages 104–109, June 1989.