himalayan linguistics

A free refereed web journal and archive devoted to the study of the languages of the Himalayas

Himalayan Linguistics

Namsel: An optical character recognition system for Tibetan text

Zach Rowinski **Kurt Keutzer**

University of California, Berkeley

Abstract

The use of advanced computational methods for the analysis of large corpora of electronic texts is becoming increasingly popular in humanities and social science research. Unfortunately, Tibetan Studies has lacked such a repository of electronic, searchable texts. The automated recognition of printed texts, known as Optical Character Recognition (OCR), offers a solution to this problem; however, until recently, robust OCR systems for the Tibetan language have not been available. In this paper, we introduce one new system, called Namsel, which uses Optical Character Recognition (OCR) to support the production, review, and distribution of searchable Tibetan texts at a large scale. Namsel tackles a number of challenges unique to the recognition of complex scripts such as Tibean *uchen* and has been able to achieve high accuracy rates on a wide range of machine-printed works. In this paper, we discuss the details of Tibetan OCR, how Namsel works, and the problems it is able to solve. We also discuss the collaborative work between Namsel and its partner libraries aimed at building a comprehensive database of historical and modern Tibetan works—a database that consists of more than one million pages of texts spanning over a thousand years of literary production.

Keywords

NLP, Tibetan, Optical Character Recogition, OCR

This is a contribution from *Himalayan Linguistics, Vol. 15(1): 12–30.*
ISSN 1544-7502

Himalayan Linguistics, Vol. 15(1). © Himalayan Linguistics 2016
ISSN 1544-7502

Namsel: An optical character recognition system for Tibetan text

Zach Rowinski **Kurt Keutzer**
University of California, Berkeley

1 Introduction

Advanced textual analysis tools such as n-gram modeling, topic modeling, and document clustering, require a corpus of digitally analyzable texts ("e-texts") in a format such as Unicode. Scattered efforts to manually input texts in Tibetan to create such an e-text corpus have typically focused on a narrow selection of works particular to a handful of genres, collections, and time periods. Moreover, many of these e-text collections, once created, are often difficult to locate and obtain. The automated recognition of printed texts or manuscripts, known as Optical Character Recognition, offers one promising solution to these problems. Unfortunately, due to lack of research funding and negligible commercial value, the Tibetan language has hitherto lacked such a tool. While long available for languages using Roman scripts, as well as for several major Asian languages, robust Optical Character Recognition implementations for Tibetan have been elusive.

The goal of our work has been to create an integrated platform, called Namsel, which uses Optical Character Recognition to support the production, review, and distribution of Tibetan e-texts at a large scale. In particular, through collaborations with the Tibetan Buddhist Resource Center (www.tbrc.org) and Tibetan and Himalayan Library (www.thlib.org), Namsel is helping bring Tibetan studies on par with the world's major languages by making publicly available a large e-text corpus of Tibetan works that span millions of pages and over a thousand years of literary production. In what follows, we discuss the state of Tibetan OCR, what Namsel does, and how it works. We then further describe the collaborative efforts among the Namsel project, UC Berkeley, the Tibetan Buddhist Resource Center, and the Tibetan and Himalayan Library, that are aimed at making available to the public a large set of both e-texts and research tools to advance the state-of-the-art in research in Tibetan studies.

2 Tibetan OCR: History and related work

Optical Character Recognition (OCR) is a computer-vision technology that converts text on printed pages and handwritten manuscripts into a machine-readable encoding, such as Unicode text. It is commonly used to enable users to search documents in the form of PDFs and scanned images. Large scale projects such Google Books (books.google.com) and the Hathi Trust repository (www.hathitrust.org) make use of OCR to produce digital tools for search and analysis of enormous corpora spanning hundreds of languages and hundreds of millions of pages.

While long available for a variety of popular languages, OCR for Tibetan, despite over two decades of active research, has only recently come into its own as robust technology used for the

production of e-text materials. As far as we know, the earliest work on Tibetan OCR began with a collaboration among Bell Laboratories' researchers Henry Baird and Kurt Keutzer, with University of Virginia student Reed Fossey (Baird 1990). Following this work, the next researcher to tackle this problem was Masami Kojima of the Tohoku Institute of Technology (Kojima 1995). While showing some promise, both Baird's work and Kojima's work struggled with the challenging issue of character segmentation (a topic we describe in detail later in this paper).

Various OCR implementations are reported in literature from China since at least the mid-1990s (Wei-lan 1999, Hao 2001: 41-16, Wang 2001: 93-96, Xiaoqing 2004: 5296, Ngondrup 2010). Perhaps the most comprehensive work to date is from Prof. Xiaoqing Ding and her students at Tsinghua University (Xiaoqing 2007: 73-98). In their work, they describe a full end-to-end pipeline for normalizing, segmenting, and recognizing Tibetan text. For isolated characters, they report an accuracy rate of 99.8% on a custom dataset using modified quadratic discriminant analysis for recognition. For samples requiring segmentation, they report an overall accuracy of 95.06%.

OCR for handwriting and blockprinted texts has received less attention. Kojima (2000) describes an approach using manual, hand-drawn segmentation markers to guide OCR and achieved modest results on a small set of test pages. Heming et al. report up to 97% accuracy on a database of handwritten Tibetan characters in uchen (Huang 2012: 5987–5993, Huang 2014: 1034-1037). According to our knowledge, none of these various implementations have been publicly released or are actively maintained.

Current approaches to printed Tibetan OCR include the Yakpo system developed by Vladimir Danilov and Alexander Stroganov.[1] They report an accuracy of 99% on a variety of fonts and were the first researchers to show the potential of using OCR to build a significant repository of major Buddhist works. Others have attempted to augment existing multi-language OCR programs, including Tesseract[2] and Abbyy OCR – with limited degrees of success. Google has implemented an early version Tibetan OCR for use in their Google Books and Google Drive products. Accuracy rates at or above 90% were observed in initial experiments and, while Tibetan does not appear to be a high priority for Google, work is reportedly ongoing.

As mentioned above, Keutzer first investigated OCR of Tibetan working with Henry Baird while both were at Bell Laboratories in the late 1980s. Later, at Berkeley, Keutzer advised a Masters student, Fares Hedayati, on OCR research, and their work was published in Hedayati's Masters Thesis entitled OCR of Tibetan Wood Block Print, as well as in (Fares 2011). While these efforts did not result in a usable system, his work did assist in outlining some of the major challenges to Tibetan OCR. In particular, Hedayati helped illustrate the use of Hidden Markov Models for woodblock texts, an approach that would influence subsequent work at Berkeley.

The focus of this paper is the Namsel system. Namsel began in 2011 with the efforts of Zach Rowinski, while working with David Germano on the Tibetan and Himalayan Library, and the basic approach and core capabilities of Namsel were developed at that time. In 2013, the effort moved to the University of California-Berkeley where Rowinski began work with Prof. Kurt Keutzer of the Department of Electrical Engineering and Computer Science. After moving to Berkeley, the project integrated earlier results of work of Heyadati on applying Hidden Markov Models and began an

1 http://www.dharmabook.ru/ocr/. Retrieved on January 15, 2016.

2 See, for example, https://groups.google.com/d/msg/tesseract-ocr/ONkAD2kuxUQ/EQsepM67D94J. Retrieved on January 15, 2016.

active collaboration with the Tibetan Buddhist Resource Center to produce a large corpora of Tibetan e-texts.

3 Namsel OCR

Namsel OCR's main product is searchable text rendered in universally accessible Tibetan fonts. Modern digital Tibetan on nearly all major operating system platforms encodes Tibetan according to the Unicode standard and that is what Namsel uses.[3] The Unicode encoding for Tibetan[4] specifies a total of 211 characters. These include the 30 standard consonants and 4 vowels, subjoined forms of 44 consonants, as well as numerous punctuation and specialized character symbols. We refer to a set characters that are delimited by whitespace or punctuation as a "syllable" and the individual columns of characters within a syllable as a "stack" or "morpheme cluster." Standard Tibetan orthography allows for approximately 500 individual stacks, which includes all legal combinations of prefix, suffix, and subjoined consonants paired with all possible vowel placements within a syllable.[5] When allowing for arbitrary combinations of characters, such as those found in Tibetan transliterations of Sanskrit and other languages, the number of stacks found in historical and modern literature increases to well into the thousands.[6] A sample of various character combinations is shown in Image 1.

Image 1: On the left is a Tibetan syllable containing four stacks (6 unique Unicode characters, omitting punctuation, in one of four horizontal positions in the syllable). On the right – (a) examples of standard Tibetan consonant clusters, (b) Tibetan transliterations of Sanskrit

This relatively large number of character combinations is one of several challenges to Tibetan OCR that are either not found in OCR for Roman scripts or not present in as much abundance. Other notable challenges particular to Tibetan OCR include:

- numerous morphologically similar graphemes, such as *pa* and *ba* or *ska* and *sga*
- poor, non-standardized layout of fonts and page-lines in many works, resulting in overlapping characters and lines
- novel, genre-specific notation found in particular works from the areas of medicine, music, and scriptural exegesis

3 https://en.wikipedia.org/wiki/Unicode

4 https://en.wikipedia.org/wiki/Tibetan_(Unicode_block)

5 A comprehensive list of legal character combinations can be found at Fynn, C. (n.d.). Elements of the Tibetan writing system. Retrieved January 4, 2016, from https://sites.google.com/site/chrisfynn2/home/tibetanscriptfonts/thetibetanwritingsystem

6 At present, Namsel OCR is able to recognize approximately 500 transliterated character clusters, mostly from those found in Buddhist canonical materials.

In addition, like all OCR implementations, Namsel needs to be able to handle a variety of issues related to complex formatting, poor printing and capture methods, and degraded source materials. In the next section, we further describe the Namsel system, how it works, and how it addresses some of these challenges.

3.1 Namsel components

3.1.1 Overview

Namsel OCR processes texts in a feed-forward manner. As shown in Image 2 below, OCR begins with image preprocessing, continues to various stages of page layout analysis that includes identifying the locations of lines and characters, and then finishes with the actual character recognition.

Namsel OCR is able to process two page styles – that of modern bound book, as well as an elongated page printed in the style of a traditional Tibetan *pecha* (Tib: *dpe cha*). Both are processed similarly: in the preprocessing stage, scanned images are cropped and binarized (converted to black and white). Incidental markings and "salt-and-pepper" noise originating from the printing or scanning process are removed. Low resolution images or blurry images are re-scaled to higher resolutions and sharpened. Borders and title lines are then detected and removed. Image contours corresponding to characters (i.e. all the black pixels on a white page) are then isolated and assigned to lines of a page. For each line, these character contours are organized, segmented, and scaled before being sent to a feature extractor. The feature extractor analyzes the images and extracts numerical attributes that encode details about its shape, orientation, and number of strokes used. These encoded features are then passed to the recognizer, which assigns character labels to each one.

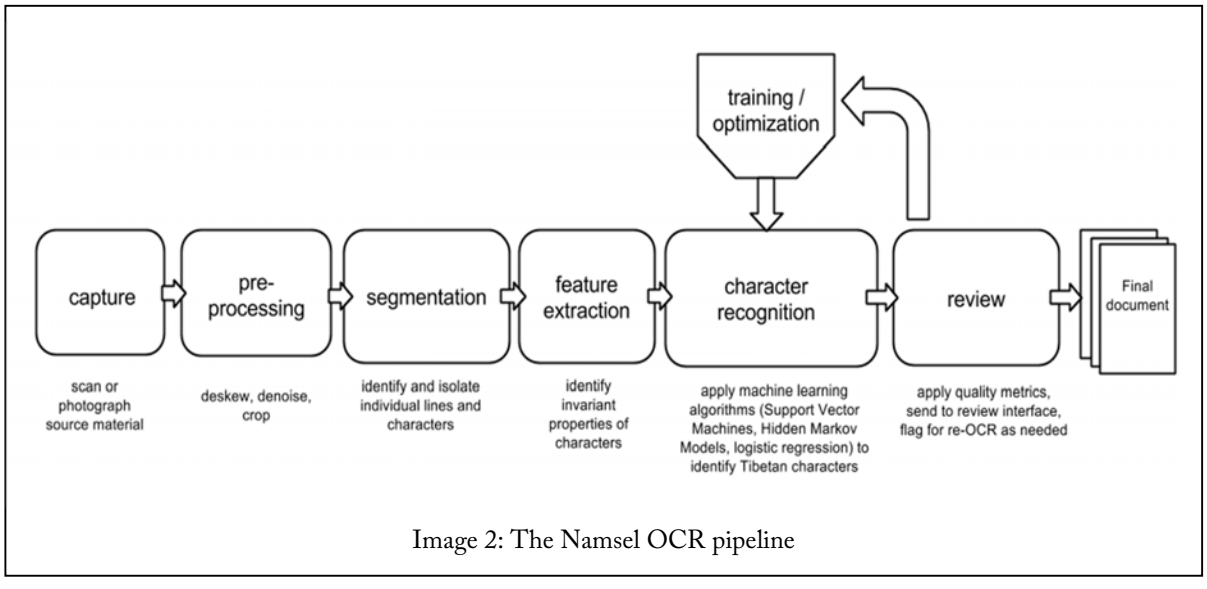

Image 2: The Namsel OCR pipeline

While broadly similar to other OCR approaches (and many computer visions pipelines generally), there are many design details unique to our implementation that are worth describing in more detail.

In what follows, we discuss some of the major components of the OCR pipeline, with a particular emphasis on the segmentation, features extraction, and recognition stages.

3.1.2 Line separation

Namsel processes a page line-by-line from left to right and top to bottom. As can be seen from Image 2, after capture and pre-processing, the next step in the process is segmentation. Here, to begin, each individual line needs to be identified and isolated. This can be a challenging task depending on, for example, how closely lines are printed together or whether or not there are incidental markings such as underlines or noise artifacts from the scanning or printing process. Closely formatted lines may contain characters that overlap across line boundaries, making them hard to separate. Noise in the form of handwritten notes, highlighting, or underlines may further complicate line separation and bring the entire recognition process to a halt. Given all the opportunities for error, robust OCR systems need to be able to employ a variety of line separation methods.

Namsel's line separation strategy makes use of one of two approaches based on the type of page being processed and its print quality: (1) naïve linear segmentation and (2) non-linear line clustering. The first approach is depicted in Image 3. Here, a horizontal density projection is created for an entire page. This projection is essentially just a sum of black or white pixels along each pixel-row of the image matrix. Linear segmentation iterates through values of the projection profile and separates lines opportunistically based on the assumption that any contiguous span of empty pixels across a page is a valid line separation path. While this assumption is often valid for very clean printings, it is far from fool-proof. Vowel characters, for example, that "float" above or below their parent consonants can easily be mis-assigned to their own or another line by virtue of the fact that there exists a straight path across the page that can cleanly pass between the vowel and the line it should be associated with. Moreover, simple linear segmentation approaches fail in the obvious case of when there is no linear path across a page between lines. While there are any number of heuristic modifications to simple linear segmentation that would help remedy this problem (some of which Namsel employs), it is often useful to opt for more complex, non-linear approaches.

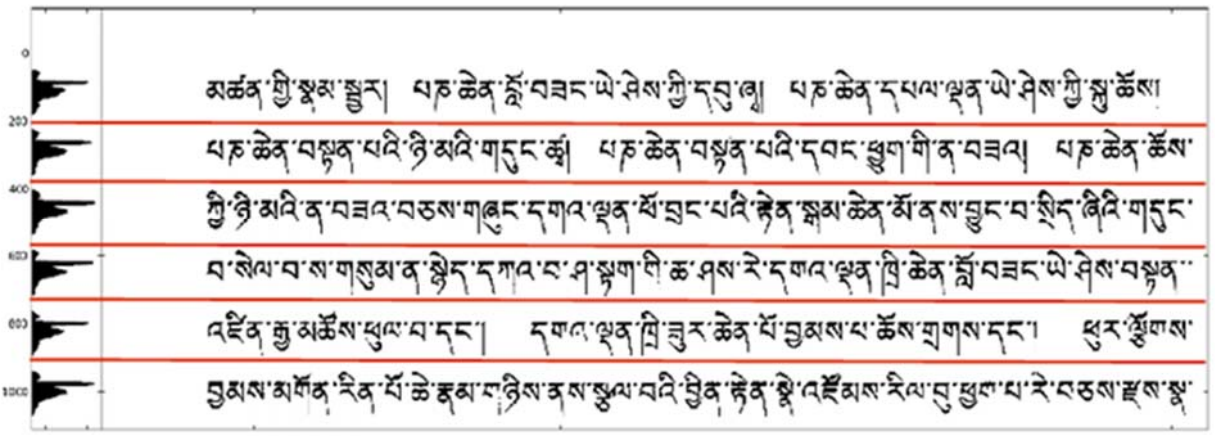

Image 3: Line separation using a projection profile. To the left, the spikes in the graph correspond to points across the page with the highest density of black pixels. The red lines indicates positions where lines can be broken apart cleanly.

Line separation via clustering is one such non-linear approach. It takes as its point of departure the assumption that the coordinate locations of all characters or character-candidates on a page have been identified and collected during preprocessing. With this data on hand, clustering involves grouping together characters based on their relative location to one another in the vertical direction using the K-Means (Hastie 2009: 509) algorithm. For this approach to be workable, the exact number of lines needs to be known prior to the assignment of characters to those lines. Namsel determines the number of lines on a page through a series of steps: first, the page image is transformed using dilation and blurring. This is intended to accentuate the location of lines and merge any floating characters to their parent lines. A horizontal projection is generated from the blurred page, which is smoothed with a Gaussian kernel and then analyzed for local minima. The total number of local minimum points (perhaps subject to some threshold) is used as the number of lines on a page. Once determined, K-Means is used to calculate a cluster center location for each line. Each character-candidate is then assigned to a line based on its distance to the nearest cluster center.

3.1.3 Character segmentation

After separating an image of text into lines, the next step is character segmentation. In machine learning tasks involving sequence prediction, such as OCR or speech recognition, recognition strategies can be classed in one of two categories: those making use of *explicit* or *implicit* segmentation. In OCR, explicit segmentation refers to the approach of identifying and extracting (or, say, drawing a box around) the portions of an image corresponding to each character (or stack) on a page. In OCR, implicit segmentation refers to an approach in which pixels of the text to be recognized are processed in a continuous stream of small strips or windows, and the strips are put together to form characters.

Unlike some Generalized Hidden Markov Model and Recurrent Neural Network approaches for OCR and speech recognition, Namsel OCR relies on an explicit rather than implicit segmentation strategy. This is a critical aspect of Namsel's design and one that has its own benefits and drawbacks. For the majority of printed Tibetan, an explicit segmentation strategy has a number of benefits: it is relatively simple to implement, it is computationally efficient, and—in many cases—it provides useful data about the size and spacing of elements on a page that can be used in formatting OCR output in a human-friendly manner. The primary drawback of explicit segmentation approaches lies in their inability to adequately model typesets and handwriting styles in which the boundaries between characters on a line are not immediately obvious (to the machine). For machine-printed Tibetan, this situation is common, albeit at varying levels of severity. Image 4 shows several examples of unsegmented characters and it is easy to see how a computer program might have difficulty segmenting this text.

To elaborate further on Namsel's approach to segmentation, we direct the reader to Image 5 below of a prototypical Tibetan syllable. Namsel's character segmentation approach would break apart the pictured syllable into four parts and OCR them individually. The four parts of the syllable include a prefix character (CbCC), the stack of the root character (EC and what lies above and below, the first-suffix (CaCC1), and second suffix (CaCC2). [7]

7 In contrast, an implicit segmentation strategy such as that used in HMM-based models might have little or no knowledge about the structure of Tibetan syllables, opting instead to create its own internal representation of the data for use in recognition.

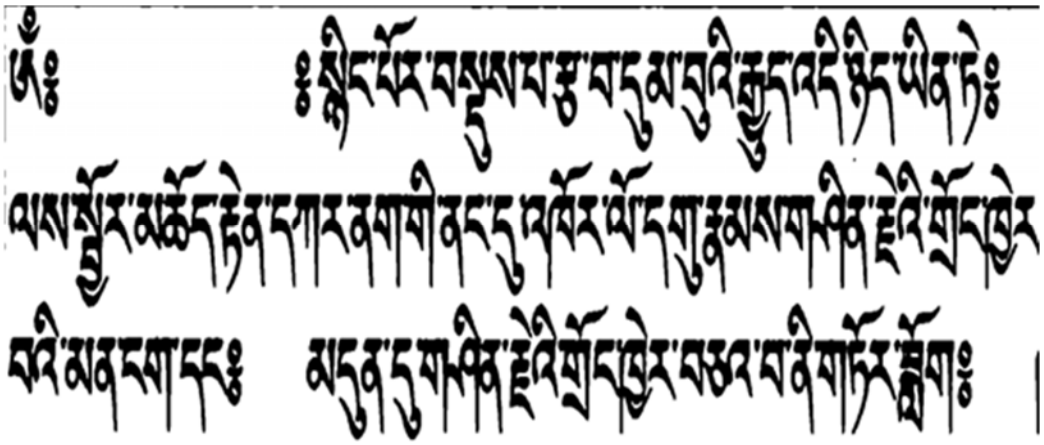

Image 4: An example of printed text with numerous horizontally touching characters that require segmentation.

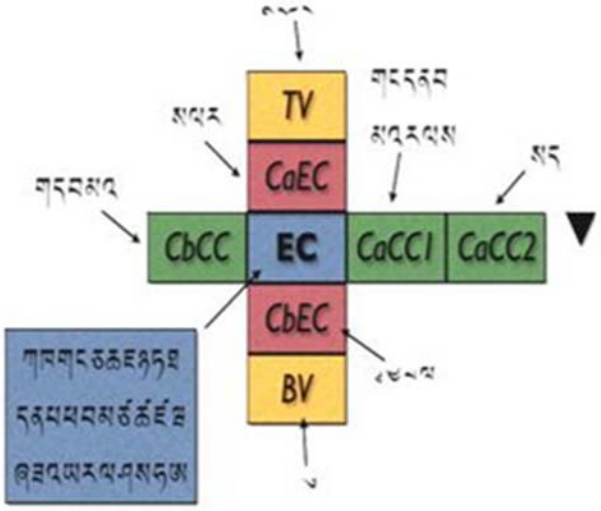

Image 5: Illustration of a Tibetan Syllable – after Masami

In order to segment attached characters, Namsel employs a probabilistic break-and-recognize approach. At a high level, this approach can be summarized as an inference process based on (1) what the system knows about Tibetan spelling and (2) *how wide* it thinks characters *should* be on any given page. Knowledge about Tibetan spelling is encoded in a transition probability matrix, while character width is encoded statistically using a Gaussian Mixture Model (GMM) with width inputs taken from the bounding box coordinates of characters/character-candidates collected in preprocessing.

The transition probability of two characters or stacks in Tibetan refers to the probability of one stack being followed by the other. Illegal character sequences, such as two consecutive Tibetan root-letters would be assigned a very low probability (close to 0), while commonly paired sequences such as pa+sa (*pas*) or do+na (*don*) would have relatively high probabilities. The transition probabilities between stacks are calculated using a large corpus of e-texts consisting of hundreds of thousands of pages. Novel character or stack pairs not present in the e-text corpus are assumed to be extremely rare and assigned probabilities close to zero. Probabilities are stored in an LxL sized matrix where L is the number of stacks that can be recognized by the system.

A Gaussian Mixture Model is a probability model that decomposes a population into one or more subpopulations, each of which can be modeled according to a Gaussian distribution. (Hastie 2009: 214). In the case of Namsel, the population in question is characters or stacks on a page and the attribute of interest is their widths. A histogram of the widths of characters and stacks on a typical page of Tibetan text shows that widths follow a bimodal distribution, which corresponds to characters with small or large widths, respectively. Small-width characters or stacks usually include punctuation symbols such as the *tseg* and *shad* characters, while large-width characters or stacks usually correspond to letters or letter-vowel columns. Assuming both character populations are normally distributed, they can be naturally modeled with Gaussian distributions. A GMM model takes as its input the width data for characters and stacks and calculates the mean and variance for each subpopulation of characters.

Namsel uses these two models—the stack-transition matrix and the width GMM—to make decisions about how each character on a page should be segmented. By and large, most characters-candidates will be accurately isolated already and will need no further segmentation. Other character candidates will be far larger than average and will need to be segmented. The exact algorithm Namsel uses for segmentation is as follows:

- First, identify and isolate characters that need special segmentation according to their width. A useful heuristic is to choose any character contour with a width greater than one or two standard deviations from the character mean width, as specified by the Gaussian Mixture Model.
- If a character-candidate's width is below the separation threshold, assume it segmented properly and continue to the next character-candidate(s)
- For each character or stack-group that needs segmentation:
    - … and for a fixed number of iterations:
        - randomly guess widths of the to-be-separated characters by sampling widths from the character-width Gaussian distribution
        - separate character-candidates according to the width guesses
        - recognize/label each separated character
        - calculate the joint probability of the sampled widths and the recognized characters using the bigram matrix and GMM models mentioned above
    - choose the segmentation points corresponding to the highest joint recognition probability

This approach in similar to that described in Ding (2007) in that it employs a progression of coarse and then fine segmentation, although it is radically different at the level of implementation. Note that Namsel adjusts the GMM distributions to avoid their being upwardly skewed by the set of yet-

to-be segmented attached stack-groups. In some cases, Namsel also removes from the GMM model large vowels that sweep the width of its parent and adjacent characters. Such vowels introduce a higher variance to the model which can make it difficult in some cases to determine the stack groups requiring segmentation. The output of character segmentation is a set of dimensions <x, y, width, height> that give the bounding box of the segmented characters in the textual image. These segmented characters then must be recognized as individual Tibetan characters.

3.1.4 Feature extraction

As illustrated in Image 2, feature extraction occurs immediately before character recognition. While the stroke pattern for any given character in Tibetan is typically the same across instantiations of that character, due to printing and font differences, small variations do exist in shape, stroke width, size, and so on from letter to letter. Robust OCR systems need to be able to accurately recognize text in spite of these differences. In machine learning and OCR, *feature extraction* is the task of identifying and encoding the details of the printed character samples that are invariant across scale, rotation, skew, small noise, and so on. Invariant features of printed text can include things like stroke patterns, the relation of character strokes to one another, and the angles of strokes. Along these lines, Namsel analyzes images for three types of invariant features: gradient directions, Zernike moments, and stroke crossings. While a detailed mathematical explanation of each of these features is beyond the scope of this paper, it is worth at least describing each briefly given their central importance to the recognition process.

Gradient directional features

Gradient directional features are derived from changes in pixel intensity across an image and are particularly useful for detecting stroke direction and edges.[8] Namsel uses a Sobel operator to obtain the gradient in the horizontal and vertical directions of an image and then combines them to generate vector directions and magnitudes across every point in an image. Namsel then divides images into 4x4 grids and registers counts of how often the gradient is pointing in one of twelve directions within each grid. The resulting output is a 192-length list of counts (12 directions x 16 grids) that effectively encodes the local stroke behavior of characters across an image.

8 See Cheriet 2007: 59-60. As implemented in Namsel OCR, gradient direction features are similar to the so-called Histogram of Oriented Gradients or HOG features, popular in computer vision applications.

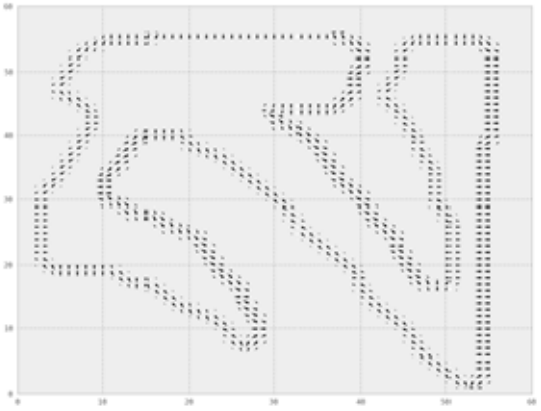

Image 6: Gradient transform of the letter *sa*. The arrows following the contour of the character are pointing in the direction of the largest change in pixel intensity.

Zernike moments

Image moments encode global properties of an image such as shape area and center of mass. Zernike moments are a particular type of image moment where, formally stated, the image pixel values are projected onto the complex conjugate of the Zernike polynomial. In image and signal processing, Zernike polynomials have a useful quality of being orthogonal, which means that individual Zernike moment features do not encode redundant properties about an image. Moreover, Zernike moments are invariant to rotation, as shown in the image below, and thus assist in making Namsel robust to skewed and slanted letters.[9]

No rotation: 0.11004072, 0.05363811, 0.19258584, 0.02261489, 0.13127123, …
90 deg CW: 0.11004072, 0.05363811, 0.19258584, 0.02261489, 0.13127123, …
Image 7: Moment values are identifical for a character at different rotations

Stroke crossings

Finally, Namsel also uses as features a count of stroke crossings made by lines projected horizontally and vertically across an image, as shown in the image below. Namsel records crossings along 7 positions in both directions, making for a total of 14 count features.

9 Namsel computes Zernike moments following Hosny, K. M. 2007.

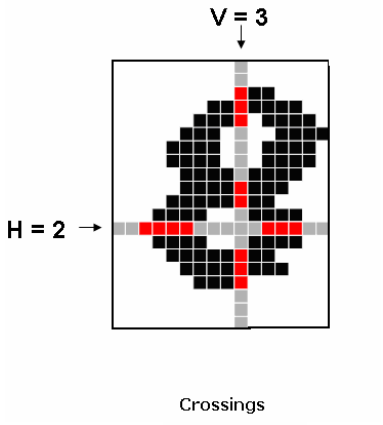

Image 8: Stroke crossing in the vertical and horizontal directions. Image from Vamvakas (No date: slide 21).

The table below summarizes the different features used by Namsel and the type of invariance they encode.

Feature Class	# Extracted	Features Extracted	Invariance
Gradient directional features	192 (12 direction bins x 16 image sections)	Counts of gradient angles	Shift Line thickness Scale
Zernike moments	90	Absolute value of Zernike moments to degree 17	Shift Rotation
Crossings	14	Horizontal and vertical crossings	Shift Rotation Line thickness

Table 1: A summary of the types of features extracted by Namsel and the invariance they encode.

At recognition time, all the feature values are concatenated together for each character and sent to the recognizer for classification.[10]

3.1.5 Character recognition

Continuing in the pipeline shown in Image 2, the next step in the OCR process is the actual character recognition. The task at this stage of the pipeline is to assign a correct label to all

10 A final note about feature extraction: handwritten features such as Zernike moments and gradient directions are not strictly required for attaining accurate OCR and are arguably less than ideal given the difficulty of engineering feature sets that are invariant to all types of character distortion. To be sure, early Namsel experiments trained on a single font with no feature extraction attained accuracy rates over 95% using a basic linear classifier. In multi-font experiments, however, such simple approaches show their limitations, with accuracy rate plummeting to 60% or 70%. As discussed at the end of this paper, deep neural networks including Convolutional Neural Nets (CNNs) and Recurrent Neural Networks (RNNs)—which require no feature engineering—promise to perform significantly better and the Namsel project is actively exploring their use for its second generation recognizer.

the data in a sample. In this case, the sample data are the isolated characters, as represented by their extracted features, and the labels are the final Unicode character assignments. Depending on its runtime settings, Namsel may make use of any one of a handful of character recognition methods. Typically, a "first-pass" classifier will attempt to recognize all the letters on a page. Later, after all characters have been entirely segmented, the classifier will make a second pass and, optionally, work in tandem with other classifiers to optimize classification accuracy.

The first and second-pass classifiers make use of a multiclass logistic regression model. This model takes as its input the extracted features for each character-candidate and generates a probability score over all possible character labels that sample may take. These probabilities are then either used to determine a final classification or are used with a Viterbi decoder. Optionally, low probability predictions can be re-classified with a Support Vector Machine (using a radial basis function kernel, RBF-SVM) (Hastie 2009: 417). The RBF-SVM is typically more accurate than logistic regression, but slower to evaluate and generally less accurate than logistic regression and Viterbi decoding combined.

Viterbi decoding is a dynamic programming algorithm that finds the most likely sequence of hidden states (in this case characters or stacks) corresponding to some set of observed data (image pixels). The general idea of this approach is that a set of "hidden" states or characters "generate" the observations of characters in an image. Given some model of that generation process, we can infer those hidden states (characters or stacks) from the image pixels and, in doing so, have recognized the text in the image.

Following the nomenclature of Hidden Markov Models and the Viterbi algorithm (Rabiner 1989), we refer to the *emission probability* as the probability that hidden state *i* generated an observation at point *t* in a sequence. All of these probabilities are calculated and stored in an LxM matrix where L is the number of possible hidden states (characters or stacks) and M is the number of items in the sequence (in this case, stacks or characters in a line or syllable). *Transition probabilities* refers to the probabilities that one stack follows another in a sequence. These are also encoded in a matrix and essentially take the form of a stack bigram. The transition probability matrix is compiled offline, while the emission or output probabilities are derived during recognition.[11]

The benefit of using the Viterbi algorithm is that it allows Namsel to use a less accurate, but fast classifier (logistic regression) to generate highly accurate predictions that take into account an entire line of text at a time. Put more simply, it allows Namsel to make use of contextual knowledge to predict the most likely sequences of letters. In practice, the use of the Viterbi algorithm has the effect of "smoothing over" incorrect predictions that result in illegal stack combinations.

3.1.6 Results

Namsel's isolated character recognition rate is 99.6% on test data containing samples from the majority of the most often used typesets in publications dating from the mid-twentieth century to the present. For real-world data requiring segmentation, Namsel's recognition rates usually range anywhere from 95-99% depending on the quality of a printing and the typeset it uses. For example, on the several hundreds of volumes of the recensions produced by the Pedurma

11 It is beyond the scope of this paper to describe the Viterbi algorithm in detail. For a concise introduction, see https://en.wikipedia.org/wiki/Viterbi_algorithm. (Retrieved January 25, 2016)

Publishing (*dpe bsdur khang*), which include the Tibetan Buddhist Kanjur (*bka' 'gyur*), Tengyur (*bstan 'gyur*), Namsel achieves an accuracy rate of over 99% (ignoring special markers such as footnotes and assuming the original documents are scanned at a high resolution). For other works from Pedurma which use another popular typeset, such as Taranatha's collected works, accuracy can ranges anywhere from 97 to over 99% depending on the content. Texts with large amounts of Sanskrit transliterations in Taranatha, for example, contain more OCR errors. For the many recent publications from the People's Publishing Houses of China (*mi rigs dpe skrung khang*), we are similarly able to achieve rates of 98-99%. For recent publications in *pecha* format, such as the *bka' ma shin tu rgyas pa* published at Sichuan Minorities Publishing House, we estimate we achieve anywhere from 97-99%, with numerous errors resulting from complex transliterated syllables and closely formatted/overlapping lines.

Namsel's accuracy is dependent on a wide range of factors, the most significant of which relate to typeset layout and image quality. Typesets that tightly group together all the characters on a page can completely disrupt a system like Namsel that (currently) relies on the ability to accurately model the widths of individual characters in order to properly perform segmentation. Similarly, poor image quality, either due to poorly inked printings or badly scanned source materials also adversely affect the page and layout modeling steps, making it difficult even to identify lines on let alone attempt accurate recognition.

As previously mentioned, other factors that affect accuracy include the presence of novel transliterations and uncommon symbols. Buddhist tantric literature, for example, is littered with mantric syllables of all varieties, with some of those syllables stacking letters 5 or 6 characters deep in a single column of text. Liturgical texts that mix standard Tibetan with musical notation and transliterated Sanskrit present similar challenges. In short, the "long-tail" of strange and unusual text and symbols (including text from non-Tibetan languages) found throughout Tibetan literature poses a constant challenge to Tibetan OCR and one that will take many years and a large amount of training data to handle effectively.

Training Data

Both the SVM and logistic regression models are trained with a dataset of roughly 45,000 characters. One-fifth of this data was generated synthetically using a set of ten Unicode Tibetan fonts. The remaining samples are hand labeled from a selection of over 1000 scanned page images from hundreds of published works. Prior to feature extraction and recognition, each sample is scaled and normalized to a 32x32 sized image.[12]

Results Assessment

The quality of OCR output is scored on a page-by-page basis using the ratio of "valid" syllables to the total number of syllables that appear on the page. Here, a valid syllable is one that either follows the rules of Tibetan spelling or is found in a pre-compiled list of accepted non-standard syllables (e.g. Sanskrit transliterations). This is an imperfect metric since it only accounts for OCR errors resulting in invalid syllables and not errors that result in valid, but misrecognized syllables. Language models, such as n-grams, are also used to assess the relative likelihood of predicted sequences. Using either metric, Namsel has the option to decide whether to automatically re-OCR portions of text in order to improve accuracy and/or flag them for manual review.

12 For a discussion of normalization methods, see Cheriet 2007: 36-38.

4 Namsel software

Namsel is written in a combination of the Python and C programming languages. Altogether, it has three components: (1) the OCR engine, (2) an online database of all generated OCR, and (3) an online interface for viewing and editing OCR and for correcting mistakes or re-training the OCR engine. The OCR engine consists of approximately 15,000 lines of code, two-thirds of which are Python. Namsel also makes use of the popular Scikit-Learn (Pedregosa 2011) and OpenCV (2016) machine learning and computer vision libraries. The database and online interface make use of a variety of technologies including the Django web framework, MYSQL, and the Solr search engine.

While it is our goal to release Namsel code as open source eventually, the primary focus is assisting in the development of a comprehensive corpus of Tibetan e-texts in collaboration with our library partners. Given our emphasis on the large-scale production of e-texts, Namsel in its present form is not instrumented for general use and is continually undergoing change as part of an iterated development process aimed at increasing its accuracy, robustness, and usability. As the project progresses, we will continue to provide updates about the availability of both the content we help produce and the tools we develop (including the OCR engine) in public channels such as Tibetan studies email lists, conference talks, and in publication.

5 Namsel database

The long term goal of the Namsel OCR project is to make available for search and analysis the entire historical Tibetan corpus. Working in partnership with the Tibetan Buddhist Resource Center (TBRC) and the Tibetan and Himalayan Library, Namsel has so far amassed a database of 1.5 million pages of OCR text. The following is a breakdown of the major genres represented in these materials, ordered from top to bottom according to their relative volume in number of pages:

- Collected works of various authors (Tib. *gsung 'bum*)
- Buddhist canonical literature (Tib. *bka' bstan 'gyur*)
- Transmitted teachings of the Ancients (Tib. *rnying lugs bka' ma*)
- Medicine (Tib. *gso rig*)
- Miscellaneous collections (Tib. *phyogs bsgrigs*)
- Biography/hagiography (Tib. *rnam thar*)
- Modern academic journals
- History (Tib. *lo rgyus*)
- "Fruit and Path" (Buddhist) literature (Tib. *lam 'bras*)
- Tibetology (Tib. *bod rig pa*)
- Ritual works / recitations (Tib. *kha ton*)
- Revelation texts (Tib. *gter ma*)
- Buddhist histories (Tib. *chos 'byung*)
- Stories from the Gesar Epic (Tib. *ge sar gyi sgrung*)
- Tibetan grammar (Tib. *sum rtags*)
- Poetry (Tib. *snyan ngag*)
- Outlines (Tib. *dkar chag*)

In total, the OCR output from Namsel amounts to about 15% of the Tibetan Buddhist Resource Center's collection of approximately ten million scanned images. An analysis of TBRC's image database suggests that about 25% are machine-printed and thus amenable to OCR using Namsel OCR. Given large overlaps in the works found in TBRC's collection (e.g. multiple printings of the Buddhist canon), once completed, Namsel-generated e-texts will have accounted for well over half the content in the TBRC's database, machine-print or otherwise. Of the many works processed by Namsel, some highlights include the entire Buddhist Canon (Tibetan: *bka' 'gyur* and *bstan 'gyur*), the Collected Tantras of the Ancients (Tibetan: *rnying ma rgyud 'bum*), and the collected works of major religious and political figures such as Longchenpa, Tsongkhapa, and the Fifth Dalai Lama. While portions of Tibetan collections had previously been available, never before could scholars search digital versions of *entire* set of these works.

Namsel's internal database of OCR text is organized by collection—where a collection is composed of one or more volumes, and a volume consists of some number of pages. Namsel stores not only the Unicode output it generates, but also the page size, location of each character or stack on a page, and the recognition probability or confidence score for each recognized character. When possible, information about the size and location of characters is used to mimic the spacing and layout of the original scanned material. Encoded in JSON format, the database is several tens of gigabytes in size. While the Namsel database itself is not publicly accessible, the primary use of the database is to assist in the generation of OCR documents that can be exported to outside projects and organizations such as the TBRC and Tibetan and Himalayan Library (who, in turn, catalog and disseminate them publicly).

6 Using Namsel's textual database at TBRC

A large corpus of Tibetan texts may aid researchers to quickly find information and to arrive at new insights. In Tibetan Buddhist studies, for example, much of the expertise developed by experienced scholars and translators rests on an ability to understand the vast technical vocabulary of Buddhist languages in a variety of contexts. The Tibetan phrase *rjes su dran pa* (Sanskrit: *anusmṛt*), for example, may typically be glossed as "to remember, recollect," but in the context of Buddhist religious doctrine may refer more technically to a family of meditation practices. Until now, the expertise to accurately identify nuanced usages in context has required years of dedicated effort to develop. Giving readers the ability to perform contextual searches on large corpora of canonical and exegetical texts promises to hasten the process of acquiring and deepening the understanding of novice and advanced scholars alike.

In addition to providing significantly increased contextualization for technical vocabularies, a large searchable corpus is useful in uncovering numerous instances of otherwise obscure terms. It is a common experience that a scholar will encounter an obscure word in a key passage of text that defies easy interpretation and will have no recourse but to hope he or she encounters it again in other contexts before developing a confidence in its meaning. In some cases, even over a lifetime of research, one might have only encountered a term—unknown in any lexicographic reference work—only once or twice. With large, searchable corpora, a search may now furnish dozens of instances of this term and greatly increase the chances of finding a clear definition.

As one anecdote of the power of contextual search with a large corpus from one of the primary authors of this paper (Keutzer): unless conversant with literature of the *rdzogs chen* tradition of Tibetan Buddhism, an individual might only rarely encounter the uncommon phrase *la bzla ba.*

While the phrase can literally be glossed as "to cross over a pass," it also has other shades of meaning that are often opaque to novice readers and even sometimes a mystery among seasoned scholars. Just over two years ago, after our first round of entering e-texts from OCR, we were excited about finding more than a dozen results through a search for the Tibetan phrase *la bzla ba*, many of which alluded to the connotation of "to transcend" or "resolve"." Only two years later, a search for the same phrase yields over 200 matches over the corpus of the famous author Longchenpa alone. The next frontier is to develop tools to organize our embarrassment of riches.

7 Future directions

Attacking blockprint or improving accuracy on texts produced by modern methods?

For decades, it appeared that OCR on any texts besides traditional xylographs or handwritten manuscripts would be of little use given the paucity of works printed in modern typesets. This posed a considerable challenge to the goal of corpus development since OCR for blockprints and handwriting is very much an unsolved problem in computer vision due to numerous difficulties. As discussed in Hedayati (2001), these difficulties include the high variation among glyphs, hard-to-isolate lines, and characters that require advanced segmentation. However, due to the publishing boom of Tibetan works within China and elsewhere, vast amounts of the extant works in the Tibetan language either have been or are in the process of being re-published using modern printing methods. For this reason, the Namsel project is focusing our efforts on improving accuracy on texts produced by modern methods rather than tackling blockprint OCR.

Such an observation leads to a common question: if the majority of texts you are able to OCR have been produced by computer or other modern typesetting methods, wouldn't it be easier to simply obtain the original computer input? The answer is, unfortunately, "no." By and large, it has been our experience and the experience of the libraries we collaborate with that the original electronic documents for input texts are unavailable and presumed lost (particularly for printed materials generated a decade or more ago).

Using state-of-the-art textual analysis tools on the e-text corpora

Robust optical character recognition is only part of Namsel's broader mission. In the long term, the goal of Namsel research and development is to provide both content and tools that enable powerful analytic capabilities to scholars of Tibetan history, language, and culture. Some areas of interest include search, language modeling, topic modeling, document clustering, intertextuality, word segmentation, and word/document embedding. We are also interested in working with linguists in support of natural language processing research on Tibetan, including, but not limited to, the development of new part-of-speech taggers.

What follows is a brief summary of work we have pursued so far. As noted above, naively searching a large e-text corpus can give an unmanageable number of matches, so improving search has been a top priority. In particular in the area of search, we are working with the Tibetan Buddhist Resource Center to explore ways to enhance the quality of search results for their e-text corpus. In the area of topic modeling, we have conducted several illuminating (and unpublished) experiments analyzing Tibetan news articles and are actively in the process of preparing major collections of OCR works data for further experimentation. Another exciting area is word-embedding (Mikolov 2013), which uses neural networks to encode the semantic relationships between words in a corpus. Many

of these methods rely on high volumes of digital texts to generate reliable results and thus stand to benefit greatly from Namsel's sizeable corpus.

Applying the state-of-the-art in machine learning to Tibetan OCR

On the OCR front, Namsel's immediate focus is to tackle the following tasks related to accuracy and content development:

- better handling of hard-to-model text, i.e. text with widely varying character widths/predominance of touching characters
- better handling of multiple languages and scripts within a text
- the acquisition of more training data of novel symbols and transliterations
- higher accuracy on poorly inked and low resolution images
- more robust page layout analysis that better handles texts with complex page structure, including images
- the development of new interfaces for editing and extracting structured data from OCR text

In this direction, we anticipate a major piece of the next generation of Namsel to rely on new state-of-the-art techniques coming from the fields of machine learning and computer vision, particularly neural networks. After recent advances (Krizhevsky 2012), neural networks (Lecun 1989, Rummelhart 1988) have proved to be powerful tools for solving a variety of naturally occurring problems in computer vision, including Optical Character Recognition of non-Roman scripts. In particular, following on the work of Graves (2007), Thomas Breuel and his students have shown Long-Short Term Memory (LSTM) machines, a particular variety of Deep Neural Nets, to be useful for Sanskrit OCR (Karayil 2015). With Breuel's generous help, the use of his open source CLSTM package (Breuel 2016), and our own ample supply of training data, we were able to perform a variety of experiments on this approach. While we were quickly able to achieve 95% accuracy on printed examples, our results stalled at this level. Similar results are reported by Karayil on Sanskrit (2015). Note that these are much lower accuracy figures than we name in our results section above. It appears that the Tibetan language's complex stacking of characters (See Image 5) confounds the direct application of (1-D, bidirectional) LSTM (as the implemented in the CLSTM project). That said, we believe that successful application of LSTM-based approaches awaits further evolution of the underlying technology and anticipate its use in future Namsel pipelines.

8 Conclusion

In this paper, we have discussed the historical development and the current state-of-the-art of Tibetan OCR, and introduced Namsel, an integrated platform for generating and disseminating Tibetan electronic texts (e-texts). We outlined the basics of how Namsel works, the problems it solves, how it solves them, and highlighted a number of areas where it can be improved. We discussed the Namsel-TBRC collaboration on building a comprehensive corpus of historical Tibetan literature and various ways it is already being used to advance scholarship in Tibetan language, history, and culture. Finally, we look forward to new opportunities on the near-horizon in the realm of text analytics and NLP afforded by the availability of a large Tibetan corpus.

We consider this an exciting time in the realm of Tibetan information technology and are thrilled to be part of a community scholars in text analytics, NLP, the humanities, and beyond

interested in promoting Himalayan languages and advancing humanity's understanding of their history, development, and use..

References

Baird, Henry S., Fossey, Henry S., and P. Lofting. 1990. "The typestyle jockey: Putting the horse out front in Devanagari and Tibetan". In: *Nordic Institute of Asian Studies report*, 5-30. Copenhagen.

Breuel, Thomas, CLSTM: github.com/tmbdev/clstm. Accessed January 25, 2016.

Cheriet, Mohamed; Kharma, Nawwaf; Lio, Cheng-Lin; Suen, Ching Y. 2007. *Character recognition systems: A guide for students and practitioners*. Hoboken, N.J: John Wiley & Sons.

Ding, Xiaoqing; and Wang, Hua. 2004. "New statistical method for multi-font printed Tibetan/English OCR." *Proceedings of SPIE-IS&T electronic imaging*, SPIE: 5296.

Ding, Xiaoqing; and Wang, Hua. 2007. "Multi-font printed Tibetan OCR." In: Chaudhuri, Bidyut (ed.) *Digital document processing*, 73-98. London: Springer.

Graves, Alex; Fernández, Santiago; Liwicki, Marcus; Bunke, Horst; and Schmidhuber, Jeurgen. 2007. "Unconstrained online handwriting recognition with recurrent neural networks." NIPS, Vancouver, Canada. http://www.cs.toronto.edu/~graves/nips_2007.pdf. Accessed June 28, 2016.

Hastie, Trevor; Tibshirani, Robert; and Friedman, Jerome. 2009. *The elements of statistical learning*. New York: Springer.

H.M. Huang, F.P. Da. 2012. "A database for off-line handwritten Tibetan character recognition," *Journal of Computational Information Systems* 9.18: 5987-5993.

H.M. Huang, F.P. Da. 2014. "Sparse representation-based classification algorithm for optical Tibetan character recognition", *Optik - International Journal for Light and Electron Optics* 125.3: 1034-1037.

Wang, Hao Jun; Zhao, Nan Yuan; and Deng, Gang Yi. 2001. "A stroke segment extraction algorithm for Tibetan character recognition (in Chinese)." *Journal of Chinese Information Processing* 15.4: 41-46.

Hedayati, Fares; Chong, Jike; and Keutzer, Kurt. 2011. "Recognition of Tibetan Wood Block Prints with generalized hidden markov and kernelized modified quadratic distance function," *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*. ACM.

Hosny, K. M. 2007. "Fast computation of accurate Zernike moments." *Journal of Real-Time Image Processing* 3.1-2: 97–107.

Kang, C., Jiang, D., and Dai, Y. 2004. "A recognition algorithm of Tibetan based on components (in Chinese)." *Proceedings of Symposium on Chinese Minority Language Information Processing and Language Resource Construction*.

Karayil, Tushar; Ul-Hasan, Adnan; and Breuel, Thomas M. 2015. "A segmentation-free approach for printed Devanagari script recognition." In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference*. IEEE: 946-950.

Kojima, Masami; Yoshiyuki Kawazoe; and Kimura, Masayuki. 2000. "A study of character recognition for Wooden Blocked Tibetan manuscript." http://pnclink.org/annual/annual2000/2000pdf/6-5-1.pdf. Retrieved January 20, 2016.

Krizhevsky, Alex; Sutskever, Ilya; and Hinton, Geoffrey E. 2012. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems*, pp. 1097-1105. Cambridge, Mass.: MIT Press.

Masami, K.; Yoshiyuki, K.; and Masayuki, K. 1996. "Character recognition of wooden blocked Tibetan similar manuscripts by using Euclidean distance with deferential weight." *IPSJ SIGNotes Computer and Humanities* 30: 13-18.

Mikolov, Tomas, et al. 2013. "Distributed representations of words and phrases and their compositionality." In: *Advances in neural information processing systems*, 3111-3119. Cambridge, Mass.: MIT Press.

LeCun, Yann, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. 1989. "Backpropagation applied to handwritten zip code recognition." *Neural Computation* 1.4: 541-551.

Ngodrup et al. 2010. "Study on printed Tibetan character recognition." *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*. Vol. 1. IEEE: 280-285.

OpenCV: github.com/Itseez/opencv. Accessed January 26, 2016.

Pedregosa. Fabian. 2011. "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*12: 2825-2830.

Rabiner, Lawrence R. 1989. "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceeings of the IEEE* 77.2: 257-286.

Rumelhart, David E.; McClelland, James L.; and PDP Research Group. 1988. *Parallel distributed processing*. Vol. 1. Cambridge, MA: MIT press.

Vamvakas, G. "Optical Character Recognition for Handwritten Characters" (no date), ppt, [Online]. Available: http://www.slideshare.net/lovebot/off-line-handwritten-optical-character-regonisation-ocr (sic). Accessed in January 25, 2016.

Wang, Haojun; Zhao, Nanyuan; Deng, Gangyi. 2001. "A Preprocessing Algorithm for Tibetan Character Recognition." *Computer Engineering* 27.09: 93-96.

Wang, Wei-lan,. 1999. "Algorithm study on feature extracting of Tibetan character recognition". *Journal of Northwest Minorities University (Natural Science Edition)* 3: 4.

Zack Rowinski
zach@eecs.berkeley.edu

Kurt Keutzer
keutzer@berkeley.edu