Text Detectionand Character Recognitionin Scene Images with Unsupervised Feature Learning

Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J. Wu, Andrew Y. Ng
Computer Science Department
Stanford University
353 Serra Mall
Stanford, CA 94305 USA {acoates,blakec,cbcase,ssanjeev,bipins,twangcat,dwu4,ang}@cs.stanford.edu

Аннотация. Чтение текста с фотографий - сложная проблема, которая получила значительное внимание. Двумя основными компонентами большинства систем являются (i) обнаружение текста из изображений и (ii) распознавание символов, а также предложены способы создания улучшенных представлений функций и моделей для обоих. В этой статье мы применяем недавно разработанные методы машинного обучения, в частности, широкомасштабные алгоритмы для автоматического изучения признаков из немаркированных данных, и показывают, что они позволяют нам создавать высокоэффективные классификаторы для обнаружения и распознавания, которые будут использоваться в высоком точность сквозной системы. Ключевые слова - Надежное чтение, распознавание символов, изучение объектов, фото ОС.

1. Вступление

Обнаружение текста и идентификация символов на сцене изображения - сложная задача визуального распознавания. С позиции компьютерного зрения, по большей части, сложность этих изображений была связана с ручным проектированием функции [1], [2], [3] и модели, которые включают различные части предшествующих знаний высокого уровня [4], [5]. В этой статье мы получаем результаты из системы, которая пытается изучать необходимые функции непосредственно из данных как альтернатива использованию специально разработанных текстовых функций или моделей. Среди наших результатов мы достигаем самый известный в значении ICDAR 2003 набор данных.

В отличие от более классических проблем с OCR (optical character recognition), где символы обычно монотонны на фиксированных фоновых изображениях, распознавание символов в изображениях сцены потенциально более сложны из-за множества возможных вариаций фона, освещения, текстуры и шрифта. В результате, создание полной системы для этих сценариев требует от нас создание представлений, которые учитывали бы все эти вариации. Действительно, значительные усилия тратятся на создание таких систем, причем лучшие исполнения объединяют десятки умных комбинированных функций и этапов обработки [5]. Однако последняя работа по компьютерному обучению стремилась создать алгоритмы, которые могут изучать представление данных более высокого уровня автоматически для многих задач. Такие системы могут быть особенно полезны, когда необходимы специальные функции, которые тяжело создать вручную. Другая потенциальная сила из этих подходов состоит в том, что мы можем легко генерировать большие количество функций, которые позволяют повысить производительность полученных алгоритмами классификации. В этой статье мы применим одну такую систему обучения для определения того, насколько эти алгоритмы могут быть полезны при обнаружении текста на сцене изображения и распознавание символов.

Функциональные алгоритмы обучения пользовались успехом и в других областях (например, достижение высокой производительности в визуальном распознавании [6] и распознавании звука [7]). К сожалению, их большой недостаток состоит в том, что эти системы часто были слишком дорогостоящими, особенно для приложений для больших изображений. Для применения этих алгоритмов в приложении для текстовых сцен, мы будем использовать более масштабируемую функциональную систему обучения. В частности, мы используем вариант K-состояний кластеризации для обучения набора функций, аналогично системе в [8]. Вооружившись этим инструментом, мы представим результаты, показывающие влияние на эффективность распознавания при увеличении количества обученных функций. Наши результаты покажут, что это можно сделать достаточно просто, обучив множество функции из данных. Наш подход выделяется на фоне других работ в приложениях текстовых сцен, поскольку ни одна из функций, используемых здесь не были специально созданы для приложения. В самом деле, система внимательно следит за тем, что было предложено в [8].

Эта статья организованна следующим образом. Сначала мы рассмотрим некоторые работы, связанные с распознаванием текстовых сцен, а также результаты машинного обучения и компьютерного зрения, которые описаны в разделе II. Затем мы опишем обучение архитектуры, используемой в наших экспериментах в разделе III, и представим результаты наших экспериментов в разделе IV, за которым следуют наши выводы.

2. Проделанная работа

Распознавание текстовых сцен вызвало значительный интерес многих отраслей научных исследований. Хотя сейчас стало возможным

достижение чрезвычайно высокой производительности по таким задачам, как распознавание цифр в контролируемых настройках [9], задача обнаружения и выделения символов в сложных сценах остается актуальной темой исследования. Однако многие из методов, используемых для обнаружения текстовых сцен и распознавания символов, основанные на продвинутых инженерных системах, специфичны для новой задачи. Например, решения для обнаружения текста варьировались от простых готовых классификаторов, обучающихся по ручному кодированию [10] до многоступенчатых источников данных, объединяющих многие различные алгоритмы [11], [5]. Общие функции включают граничные функции, дескрипторы текстур и контексты фигур [1]. Между тем, были применены различные варианты вероятностной модели [4], [12], [13], сворачивающие многие формы предшествующих достижений в этой области в систему обнаружения и распознавания.

С другой стороны, некоторые системы с высокой гибкостью обучения пытаются получить всю необходимую информацию от помеченных данных с минимальным наличием предварительной информации. Например, архитектуры многоуровневых нейронных сетей применяются для распознавания символов и могут конкурировать с другими ведущими методами [14]. Успех такого подхода подтверждается системами распознавания текста большинства традиционных документов и рукописных текстов [15]. Действительно, метод, используемый в нашей системе связан со сверточными нейронными сетями. Основное различие заключается в том, что используемый здесь метод обучения не контролируется и использует гораздо более масштабируемый алгоритм обучения, который может быстро обучать множество функций.

На методах функционального обучения в целом в настоящее время сфокусировано множество исследований, особенно применительно к проблеме компьютерного зрения. В результате, стал доступным широкий спектр алгоритмов для обучения функций из немаркированных данных [16], [17], [18], [19], [20]. Множество результатов, полученные помощью систем функционального обучения показали, что более высокая производительность в задачах распознавания может быть достигнута посредством более масштабных представлений, которые могут быть сгенерированы масштабируемыми системами функционального обучения. Например, Van Gemert в работе [21] показал, что производительность может расти с ростом количества функций нижнего слоя, и Li в работе [22] предоставил доказательства аналогичного явления для функций высшего слоя, таких как объекты и части. В этой работе мы уделяем особое внимание подготовке низкоуровневых функции, но более сложные методы обучения способны изучать конструкции более высокого уровня, которые могут быть еще более эффективны [23], [7], [17], [6].

3. Архитектура обучения

Теперь мы опишем архитектуру, используемую для изучения функций представления и обучение классификаторов, используемых для нашей системы обнаружения и распознавания символов. Базовая установка тесно связана со сверточной нейронной сетью [15], но благодаря ее методу обучения можно быстро создавать чрезвычайно большие наборы функций с минимальной настройкой.

Наша система работает в несколько этапов:

1) Применение неконтролируемого алгоритма обучения к набору патчей изображений, собранных из данных обучения, чтобы обучить набор функций.

2) Оценить возможность свертки над изображениями обучающей выборки. Уменьшить количество функций, используя пространственные объединения [15].

3) Обучить линейный классификатор для обнаружения текста или распознавание символов.

Ниже мы опишем каждый из этих этапов более подробно.

A. Обучение функций

Основным компонентом нашей системы является применение неконтролируемого алгоритма обучения для генерации функций, используемых для классификации. Для этой цели доступны множество алгоритмов неконтролируемого обучения, такие как автокодеры [19], RBM [16] и разреженное кодирование [24]. Однако здесь мы используем вариант кластеризации K-средних, который, как было показано, дает результаты, сравнимые с другими методами, при этом намного проще и быстрее.

Как и многие другие схемы обучения, наша система работает, применяя общепринятые шаги:

1) Выбрать набор небольших изображений, $\tilde{x}^{(i)}$ из обучающей выборки. В нашем случае это набор изображений 8x8 в серых тонах[1], то есть $\tilde{x}^{(i)} \epsilon R^{64}$.
2) Применить простую статистическую предобработку (например, отбеливание) ко входному набору, чтобы получить новый набор данных $x^{(i)}$.
3) Запустить неконтролируемый алгоритм обучения на наборе x^{i}, чтобы построить сопоставление от входных наборов к вектору признаков, $z^{(i)} = f(x^{(i)})$.

Используемая конкретно нами система аналогична той, которая приведена в [8]. Во-первых, учитывая набор обучающих образов, мы извлекаем набор из m 8x8-пиксельных наборов для получения векторов пикселей $\tilde{x}^{(i)} \in R^{64}, i \in \{1, \dots, m\}$. Каждый вектор соответствует яркости и контрасту[2]. Затем мы отбеливаем $\tilde{x}^{(i)}$ с помощью отбеливания ZCA[3] [25], чтобы получить $x^{(i)}$. Учитывая этот побеленный банк входных векторов, мы теперь готовы изучить набор функций, которые можно оценить на таких патчах.

Для неконтролируемого этапа обучения мы используем вариант кластеризации K-средних. K-средство может быть модифицировано так, чтобы оно давало словарь $D \in R^{64*d}$ нормализованных базисных векторов. В частности, вместо того, чтобы изучать «центроиды» на основе евклидова расстояния, мы изучаем набор нормированных векторов $D^{(j)}, j \in \{1, \dots, d\}$, чтобы сформировать столбцы D, используя внутренние произведения как метрику подобия. То есть мы решаем:

$$\min_{D,S^{(i)}} \sum_i ||Ds^{(i)} - x^{(i)}||^2 \quad (1)$$

$$s.t. ||s^{(i)}||_1 = ||s^{(i)}||_\infty, \forall_i \quad (2)$$

$$||D^{(j)}||_2 = 1, \forall_j \quad (3)$$

где $x^{(i)}$ - входные примеры, а $s^{(i)}$ - соответствующие «одни горячие» кодировки[4] примеров. Как и K-means, оптимизация выполняется путем чередования минимизации по D и $s^{(i)}$. Здесь оптимальное решение для

s (i) задано D - установить s (i) k = D (k)> x (i) для k = argmaxj D (j)> x (i) и установить s (i) j = 0 для всех остальных j 6 = k.

Рисунок 1. Небольшое подмножество элементов словаря, полученных из оттенков серого, 8-на-8 пиксельных патчей изображения, извлеченных из набора данных ICDAR 2003

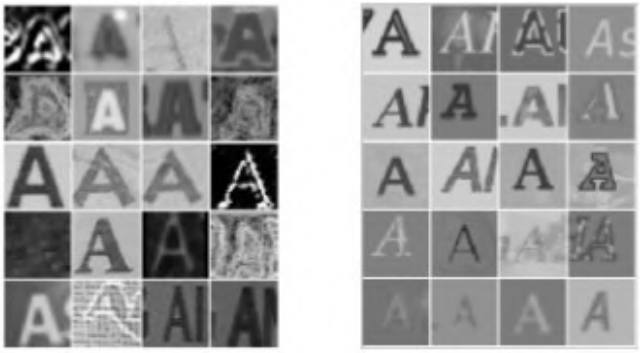
Тогда, удерживая все s (i) фиксированные, легко решить для D (в замкнутой форме для каждого столбца), а затем перенормировать столбцы. На рисунке 1 показан набор словарных элементов (столбцов D), полученных в результате этого алгоритма при применении к отбеленным пятнам, извлеченным из небольших изображений символов. Они заметно похожи на фильтры, полученные другими алгоритмами (например, [24], [25], [16]), хотя метод, который мы используем, довольно прост и очень быстр. Обратите внимание, что функции специализируются на данных - некоторые элементы соответствуют коротким, изогнутым штрихам, а не просто к краям. После того, как мы получим наш обучаемый словарь D, мы сможем затем определить представление функции для одного нового патча 8-by8. Учитывая новый входной патч ~ x, мы сначала применяем преобразование нормализации и отбеливания, используемое выше, чтобы получить x, а затем сопоставить его с новым представлением z ∈ Rd, взяв скалярное произведение с каждым элементом словаря (столбец D) и применяя скаляр нелинейная функция. В этой работе мы используем следующее отображение, которое, как мы нашли, хорошо работает в других приложениях: z = max {0, | Dx | -α}, где α - выбранный гиперпараметр. (Обычно мы используем α = 0,5.).

B. Извлечение функции

Как наш детектор, так и классификация символов рассматривают 32-х 32-пиксельные изображения. В результате представления изображения 32 на 32 мы вычисляем представление, описанное выше, для каждого суб-патча 8 на 8 ввода, что дает представление 25by-25-by-d. Формально мы будем обозначать z (ij) ∈ Rd представление паттерна 8 на 8, расположенное в позиции i, j внутри входного изображения. На этом этапе необходимо уменьшить размерность представления до классификации. Общим способом сделать это является пространственный пул [26], где мы объединяем ответы функции в нескольких местах на одну функцию. В нашей системе мы используем среднее объединение: мы суммируем векторы z (ij) над 9 блоками в сетке 3 на 3 по изображению, получая конечный вектор признаков с 9d-функциями для этого изображения.

C. Обучение текстовому детектору

Для обнаружения текста мы обучаем двоичную классификацию, которая направлена на то, чтобы различать 32-на-32 окна, содержащие текст из окна. Это должно быть сделано



(a) Искаженные примеры ICDAR (b) Синтетические примеры

Рисунок 2. Расширенные примеры обучения.

путем извлечения 32-на-32 окон из учебного набора данных ICDAR 2003, используя поле ограничения слов, чтобы определить, является ли окно текстовым или нетекстовым.5C помощью этой процедуры мы собираем набор из 60000 32-на-32 окон для (30000 положительных, 30000 отрицательных). Затем мы используем описанный выше метод извлечения признаков, чтобы преобразовать каждое изображение в 9d-мерный вектор признаков. Эти векторы признаков и метки истины «текст» и «не текст», полученные из ограничивающих прямоугольников, затем используются для обучения линейного СВМ. Позднее мы будем использовать наш экстрактор признаков и обученный классический дизайн в обычном режиме «скользящего окна».

D. Обучение классическому классу

Для классификации символов мы также используем образ размером с 32 × 32 пикселя, который применяется к изображениям символов в наборе помеченных данных поезда и тестовых данных6. Однако, поскольку мы можем производить большое количество функции, использующие вышеописанный подход к изучению функций, чрезмерная перегрузка становится серьезной проблемой при обучении из (относительно) небольших наборов данных персонажей, которые в настоящее время используются. Чтобы помочь смягчить эту проблему, мы объединили данные из нескольких источников. В частности, мы собрали наши учебные данные из учебных образов ICDAR 2003 [27], набора данных для считывания знака Weinman и др. [4] и английского подмножества набора данных Chars74k [1]. Наш комбинированный тренировочный набор содержит приблизительно 12400 обозначенных символьных изображений. С большим количеством функций полезно иметь еще больше данных. Чтобы удовлетворить эти потребности, мы также экспериментировали с синтетическими дополнениями этих наборов данных. В частности, мы добавили синтетические примеры, которые представляют собой копии учебных образцов ICDAR со случайными искажениями и применяемыми ими фильтрами изображения (см. Рис. 2 (а)), а также искусственные примеры визуализированных символов, смешанных со случайными изображениями пейзажей

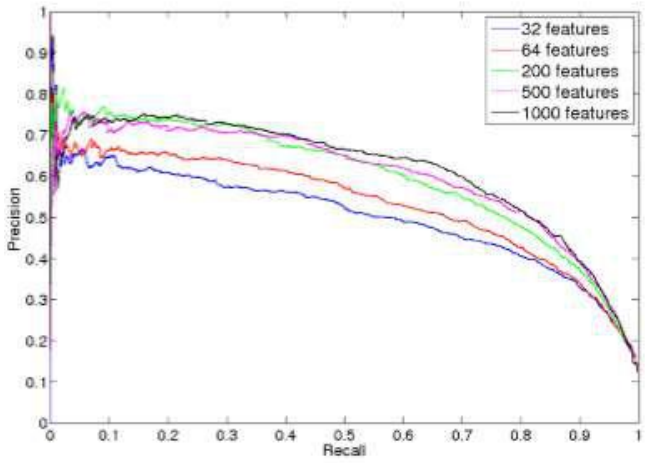


Рисунок 3. Прецизионные кривые для детекторов с различным количеством функций.

(Рисунок 2 (b)). С учетом этих примеров наш набор данных включает в общей сложности 49200 изображений.

4. Эксперименты

Теперь мы представляем экспериментальные результаты, достигнутые с помощью системы, описанной выше, демонстрируя влияние способности обучать все большее число функций. В частности, для обнаружения и распознавания символов мы обучали наших классиков с увеличением числа изученных функций и в каждом случае оценивали результаты тестов на ICDAR 2003 для обнаружения текста и распознавания символов.

A. Обнаружение

Чтобы оценить наш детектор на большом входном изображении, мы берем класс, обученный, как в разделе III-C, и вычисляем функции и классификационный вывод для каждого окна изображения 32 на 32. Мы выполняем этот процесс в нескольких масштабах, а затем для каждого места в исходном изображении присваиваем ему оценку, равную максимальной производительности класса, достигнутой в любом масштабе. По этому механизму мы помещаем каждый пиксель с оценкой в зависимости от того, является ли этот пиксель частью блока текста. Затем эти баллы порождаются для получения двоичных решений на каждом пикселе. Изменяя пороговое значение и используя ограничивающие поля ICDAR в качестве пиксельных меток, мы выводим кривую прецизионного отзыва для детектора и сообщаем область под этой кривой (AUC) в качестве нашей конечной оценки эффективности. На рисунке 3 показаны кривые прецизионного отзыва для нашего детектора для различного количества функций. Там видно, что производительность постоянно улучшается, так как мы увеличиваем количество функций. Производительность нашего детектора (площадь под каждой кривой) улучшается с 0,5 AUC, до 0,62 AUC, просто за счет включения дополнительных функций. Хотя наша производительность еще не сопоставима с наиболее эффективными системами, примечательно, что наш подход не включал практически никаких предварительных знаний. Напротив, недавняя современная система Pan & al. [5] включает в себя несколько

(a) ICDAR test image (b) Text detector scores

(c) ICDAR test image (d) Text detector scores

Рисунок 4. Пример выводов классификатора.

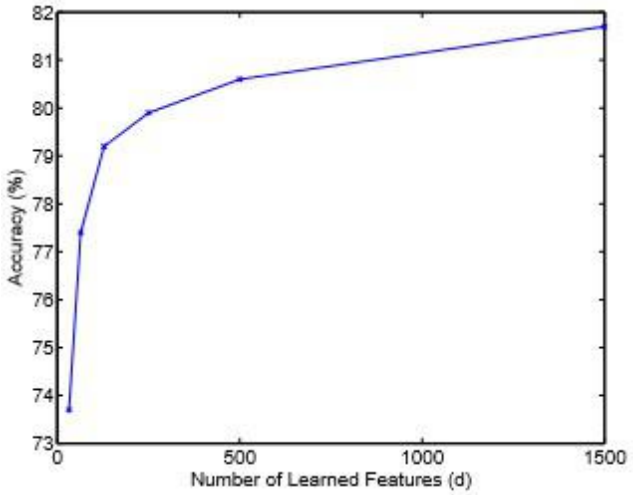


Рисунок 5. Точность классификации символов (62-way) на тестовом наборе ICDAR 2003 в зависимости от количества обучаемых функций.

высоко настраиваемые этапы обработки, включающие несколько наборов избранных экспертами функций. Обратите внимание, что эти числа имеют точность в пикселях (т. Е. Производительность детектора при идентификации для одного окна, будь то текст или не текст). На практике предсказанные метки соседних окон сильно коррелированы, и, следовательно, выходные данные включают большие смежные «комки» положительно и негативно обозначенных окон, которые могут быть переданы для большей обработки. Типичный результат, создаваемый нашим детектором, показан на рисунке 4.

B. Распознавание символов

Как и в случае с детекторами, мы обучали наших персональных классиков с различным количеством функций в комбинированном учебном наборе, описанном в разделе III. Затем мы проверили этот класс на тестовом наборе CDAR 2003, который содержит 5198 тестовых символов из 62 классов (10 цифр, 26 верхних и 26 строчных букв).

Таблица 1
ТОЧНОСТЬ ПРИЗНАНИЯ ИСПЫТАНИЙ НА ОСНОВЕ ХАРАКТЕРОВ ICDAR 2003. (DataSet-КЛАССЫ)

Algorithm	Test-62	Sample-62	Sample-36
Neumann and Matas, 2010 [28]	67.0%[7]	-	-
Yokobayashi et al., 2006 [2]	-	81.4%	-
Saidane and Garcia, 2007 [14]	-	-	84.5%
This paper	81.7%	81.4%	85.5%

Средняя точность классификации в наборе тестов ICDAR для увеличения числа функций приведена на рисунке 5. Снова мы видим, что точность поднимается как функция количества функций. Обратите внимание, что точность для самой большой системы (1500 функций) является самой высокой, на 81,7% для проблемы классификации 62way. Это сопоставимо или превосходит другие (специально созданные) системы, проверенные по одной и той же проблеме. Например, система в [2] достигает 81,4% на меньшем наборе «образец» ICDAR, где мы тоже достигаем 81,4%. Авторы [14], использующие контролируемую сверточную сеть, достигают 84,5% для этого набора данных, когда он рухнул на 36-полосную проблему (исключая чувствительность к регистру). В этом случае наша система достигает 85,5% с 1500 функциями. Эти результаты суммированы по сравнению с другими работами в Таблице I.

5. Заключение

В этой статье мы создали систему обнаружения и распознавания текста на основе алгоритма масштабируемого алгоритма обучения и применили его к изображениям текста в естественных сценах. Мы продемонстрировали, что с более крупными банками функций мы можем добиться большей точности с максимальной производительностью, сравнимой с другими системами, подобно результатам, наблюдаемым в других областях компьютерного зрения и машинного обучения. Таким образом, в то время как многие исследования были сосредоточены на разработке вручную моделей и функций, используемых в текстовых приложениях, наши результаты указывают на то, что может быть достигнута высокая производительность с использованием более автоматизированного и масштабируемого решения. Благодаря более масштабируемым и сложным алгоритмам обучения предметам, которые в настоящее время разрабатываются исследователями по компьютерному обучению, возможно, что подходы, которые здесь преследуют, могут достичь производительности, намного превышающей возможности, которые возможны с помощью других методов, которые в значительной степени зависят от предварительно знакомых вручную знаний.

ПОДТВЕРЖДЕНИЕ

Адам Коутс поддерживает стипендию Стэнфордского университета.

СПИСОК ИСТОЧНИКОВ

[1] T. E. de Campos, B. R. Babu, and M. Varma, “Character recognition in natural images,” in Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, February 2009.

[2] M. Yokobayashi and T. Wakahara, “Binarization and recognition of degraded characters using a maximum separability axis in color space and gat correlation,” in International Conference on Pattern Recognition, vol. 2, 2006, pp. 885– 888.

[3] J. J. Weinman, “Typographical features for scene text recognition,” in Proc. IAPR International Conference on Pattern Recognition, Aug. 2010, pp. 3987–3990.

[4] J. Weinman, E. Learned-Miller, and A. R. Hanson, “Scene text recognition using similarity and a lexicon with sparse belief propagation,” in Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 10, 2009.

[5] Y. Pan, X. Hou, and C. Liu, “Text localization in natural scene images based on conditional random field,” in International Conference on Document Analysis and Recognition, 2009.

[6] J. Yang, K. Yu, Y. Gong, and T. S. Huang, “Linear spatial pyramid matching using sparse coding for image classification.” in Computer Vision and Pattern Recognition, 2009.

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in International Conference on Machine Learning, 2009.

[8] A. Coates, H. Lee, and A. Y. Ng, “An analysis of single-layer networks in unsupervised feature learning,” in International Conference on Artificial Intelligence and Statistics, 2011.

[9] M. Ranzato, Y. Boureau, and Y. LeCun, “Sparse feature learning for deep belief networks,” in Neural Information Processing Systems, 2007.

[10] X. Chen and A. Yuille, “Detecting and reading text in natural scenes,” in Computer Vision and Pattern Recognition, vol. 2, 2004.

[11] Y. Pan, X. Hou, and C. Liu, “A robust system to detect and localize texts in natural scene images,” in International Workshop on Document Analysis Systems, 2008.

[12] J. J. Weinman, E. Learned-Miller, and A. R. Hanson, “A discriminative semi-markov model for robust scene text recognition,” in Proc. IAPR International Conference on Pattern Recognition, Dec. 2008.

[13] X. Fan and G. Fan, “Graphical Models for Joint Segmentation and Recognition of License Plate Characters,” IEEE Signal Processing Letters, vol. 16, no. 1, 2009.

[14] Z. Saidane and C. Garcia, “Automatic scene text recognition using a convolutional neural network,” in Workshop on Camera-Based Document Analysis and Recognition, 2007.

[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Computation, vol. 1, pp. 541–551, 1989.

[16] G. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] R. Salakhutdinov and G. E. Hinton, “Deep Boltzmann Machines,” in 12th International Conference on AI and Statistics, 2009.

[18] M. Ranzato, A. Krizhevsky, and G. E. Hinton, “Factored 3way Restricted Boltzmann Machines for Modeling Natural Images,” in 13th International Conference on AI and Statistics, 2010.

[19] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in Neural Information Processing Systems, 2006.

[20] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, “Selftaught learning: transfer learning from unlabeled data,” in 24th International Conference on Machine learning, 2007.

[21] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, “Kernel codebooks for scene categorization,” in European Conference on Computer Vision, 2008.

[22] L.-J. Li, H. Su, E. Xing, and L. Fei-Fei, “Object bank: A high-level image representation for scene classification and semantic feature sparsification,” in Advances in Neural Information Processing Systems, 2010.

[23] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, “Learning convolutional feature hierarchies for visual recognition,” in Advances in Neural Information Processing Systems, 2010.

[24] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” Nature, vol. 381, no. 6583, pp. 607–609, 1996.

[25] A. Hyvarinen and E. Oja, “Independent component analysis: algorithms and applications,” Neural networks, vol. 13, no. 4-5, pp. 411–430, 2000.

[26] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, “Learning mid-level features for recognition,” in Computer Vision and Pattern Recognition, 2010.

[27] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, “ICDAR 2003 robust reading competitions,” International Conference on Document Analysis and Recognition, 2003.

[28] L. Neumann and J. Matas, “A method for text localization and recognition in real-world images,” in Asian Conference on Computer Vision, 2010.