

Aprenentatge automàtic aplicat al reconeixement de caràcters en imatges

Francesc Aguirre,¹ Alberto Debernardi²

¹Autor: francesc.aguirre@gmail.com

²Tutor: adebernardipinos@gmail.com

23/07/2021



Aquí anirà el resum en cat, cast, ang.

Contents

1	Introducció	1
2	OCR	1
2.1	Parts tradicionals d'un sistema OCR	2
2.2	Sistema OCR a l'actualitat	4
3	Problema pràctic	5
3.1	Base de dades	5

1 Introducció

L'estadística és la disciplina que s'encarrega d'analitzar dades per a respondre a preguntes empíriques. A mitjans del segle XX, amb la creació dels dispositius electrònics d'emmagatzematge i l'ajuda de sensors, la quantitat d'informació que es pot recollir ha crescut any rere any. Aquest fet ha donat lloc al naixement de noves disciplines d'anàlisi de dades, tals com la mineria de dades i l'aprenentatge automàtic, més conegut pel seu nom en anglès *machine learning*.

El machine learning és un conjunt de tècniques que dona als ordinadors l'habilitat d'aprendre de les dades. S'utilitza per a resoldre una gran varietat de problemes predictius complexos en els àmbits d'economia i finances, bioinformàtica, salut, meteorologia, màrqueting, problemes en l'anàlisi i classificació d'imatges, vídeos, àudio... En l'àmbit de l'anàlisi d'imatges hi ha la detecció d'objectes, i més concretament, el reconeixement òptic de caràcters.

En aquest estudi s'investigarà l'àmbit del reconeixement òptic de caràcters, i com obtenir models predictius competents utilitzant tècniques de machine learning. Per a posar aquests coneixements en pràctica, es resoldrà un problema d'identificació de caràcters irregulars en imatges, tal com la identificació d'escriptura manual o la resolució de *captcha* (és a dir, text en una determinada font que inclou una certa distorsió, precisament per evitar la detecció dels caràcters per part de models més simples, i que són àmpliament utilitzats en internet per evitar l'automatització de determinats processos).

2 OCR

El reconeixement òptic de caràcters OCR (de les sigles en anglès *Optical Character Recognition*) és una aplicació de la intel·ligència artificial, que té com a objectiu detectar i identificar els caràcters que es puguin trobar en una imatge. És una de les àrees més estudiades en l'àmbit de reconeixement de patrons gràcies al gran nombre d'aplicacions pràctiques. Alguns dels prob-

lemes que OCR pot ajudar a agilitzar i automatitzar són la digitalització de diaris i llibres antics, la identificació de matrícules, la classificació d'imatges segons el text detectat, i la lectura de dades en paper tals com documentació, correu i enquestes. La digitalització a ordinador de text té moltes més aplicacions, tals com la cerca, edició i emmagatzematge d'informació, traducció, transcripció de text a àudio i NLP (de les sigles en anglès *Natural Language Processing*).

2.1 Parts tradicionals d'un sistema OCR

Tradicionalment, un problema típic d'OCR es pot dividir en subtasques en forma de *pipeline* (fetes una darrera l'altre en un ordre concret) que utilitzen tècniques de visió artificial (en anglès *computer vision*), estadístiques i de machine learning. És útil conèixer-les per saber en quin punt del sistema s'està (1).

1. Escaneig: És el procés d'escanejar o fotografiar el text *input*. Normalment, al d'escanejar un document s'aplica *thresholding* (binarització) per estalviar memòria i capacitat computacional, que és una tècnica que dicotomitza el color gris de documents en blanc i negre segons un llindar d'intensitat.
2. Segmentació de text: Un cop escanejat el document, típicament el que es vol és obtenir un sol caràcter per utilitzar-lo d'input en el model. La segmentació d'una imatge és la divisió d'aquesta en parts. En aquesta tasca, el que es vol és localitzar el text que ens interessa i ometre gràfics, imatges, logotips... Tot seguit es vol segmentar línies de text, de les línies es segmenten paraules, i de les paraules se segmenten caràcters. En la segmentació ens podem trobar diversos problemes, sobretot si un caràcter està format per diferents parts (i j), si s'està tocant amb algun altre, o s'ha dividit.
3. Preprocessament: Com que el fet d'escanejar o fotografiar una imatge és un procés variable (brillantor, angle...), els caràcters de la imatge segmentada poden contenir soroll, o

estar trencats. El preprocessament té com a objectiu eliminar soroll, omplir espais trencats i reduir la grossor dels píxels negres que formen el caràcter. També es normalitzen les dimensions i s'aplicarà una rotació si es detecta inclinació.

4. Segmentació interna: Consisteix a segmentar la imatge del caràcter en seccions més petites, tals com línies i corbes concretes. La intensió és començar a detectar zones amb patrons concrets que facilitin el reconeixement posterior del caràcter.
5. Extracció de variables: A partir de la segmentació interna, se seleccionarà un conjunt de variables que maximitzi el reconeixement amb el nombre menor d'elements. L'objectiu és capturar característiques essencials dels símbols per posteriorment entrenar el model. La extracció de variables més senzilla seria utilitzar la matriu de píxels de la imatge, tot i que utilitzar tantes variables pot provocar problemes de dimensionalitat en molts dels models. Utilitzant extracció de variables, s'utilitzarien característiques que descriuïn els caràcters, tals com llargades de segments en regions de la imatge, angles de curvatura...
6. Entrenament i reconeixement: Aplicació de tècniques de reconeixement de patrons per a classificar el caràcter. Aquí és on podem utilitzar tècniques estadístiques i de machine learning per fer la classificació, tals com (TODO)...
7. Reagrupació de caràcters a paraules, paraules a línies, fins a tenir el document complet.

Com es pot veure, la creació d'un sistema OCR tradicional és un procés llarg, amb moltes subtasques que relativament complicades. Els avantatges dels mètodes tradicionals és que donen bons resultats amb mostres petites i són computacionalment eficients. Alguns dels inconvenients és que utilitzen detecció i segmentació de text a partir de tècniques de visió artificial no relacionades amb machine learning, és a dir, no aprenen de les dades. Com que la segmentació

no sempre és evident, i les dades poden ser sorolloses, utilitzar aquest tipus de tècniques acostuma a produir errors difícils de solucionar.

2.2 Sistema OCR a l'actualitat

Aquests últims anys, a partir del 2010, el desenvolupament de la branca *deep learning* (aprenentatge profund, una branca de machine learning) a tingut un avanç molt important. La solució en les xarxes neuronals del problema de *vanishing/exploding gradients* i el llançament de noves tecnologies d'alta capacitat computacional tals com noves GPU (de les sigles en anglès *graphics processing unit*), a permès l'entrenament de xarxes neuronals profundes (DNN de les sigles en anglès *deep neural networks*).

Aquest desenvolupament ha donat lloc a nous sistemes OCR basats en xarxes neuronals. La majoria del preprocessament i binarització no és necessari, ja que les xarxes neuronals s'adapten als inputs, podent utilitzar fàcilment els píxels de les imatges. La segmentació de text es pot fer amb tècniques basades en DNN tals com FCN (de les sigles en anglès *Fully Convolutional Networks*), donant millors resultats que amb les tècniques de visió artificial. A més a més, si s'utilitzen RNN (de les sigles en anglès *Recurrent Neural Networks*), no cal segmentar caràcter a caràcter, sinó que es poden utilitzar línies completes de text. Un altre avantatge de les RNN és que poden aprendre de manera natural la llengua utilitzada en l'entrenament. La tendència actual és utilitzar FCN per la segmentació de text en línies, i utilitzar RNN pel seu reconeixement, juntament amb CNN (de les sigles en anglès *Convolutional Neural Networks*) per fer l'extracció de variables (2).

3 Problema pràctic

En aquesta secció es buscaran solucions a un problema pràctic d'OCR. L'objectiu és introduir recursos que ajudin a solucionar un problema del reconeixement de caràcters, i veure quina solució dóna millors resultats.

Inicialment es buscarà solucionar un problema de reconeixement de caràcters escrits a mà. A primera vista aquest no és un problema fàcil. Hi ha 62 caràcters diferents en l'alfabet anglès, el que implica que a partir d'una imatge, l'algoritme haurà de seleccionar un caràcter entre els 62 disponibles (classes). Si es volgués utilitzar el nostre model per automatitzar completament un procés, es buscaria una precisió propera al 99.5%. Si es té algú vigilant l'algoritme, llegint que el reconeixement tingui sentit, es pot demanar una precisió més baixa, entre 95% i 99%. A més a més, si el que s'analitza són caràcters individuals (sistema tradicional, part de segmentació de text), normalment aquest caràcter forma part d'una paraula, i si aquesta paraula no existeix en el vocabulari, podem fer que el sistema ens avisi (o per exemple, si volem llegir els NIU o id d'alumnes, podem detectar si existeix el NIU reconegut). Per tant, tot i no aconseguir la precisió més alta, el sistema pot seguir sent útil i no fer errors tan fàcilment.

3.1 Base de dades

Per a l'elecció de la base de dades, com que s'està experimentant i no es té un objectiu estricte (no s'està participant o ajudant en cap projecte extern al treball), s'ha optat per una base de dades amb una bona quantitat d'exemples i dades ja preprocessades, ja que el preprocessament de dades és una part molt tècnica i metòdica. S'han trobat diferents opcions gratuïtes disponibles en internet, i finalment s'ha utilitzat la base de dades d'imatges de caràcters EMNIST (3), derivada de la base de dades d'OCR NIST (4).

La base de dades NIST consisteix en 3669 imatges binaritzades de mostres de formularis fets expressament per testejar mètodes de reconeixement de caràcters (fig:1). La base de dades

HANDWRITING SAMPLE FORM

NAME	DATE	CITY	STATE	ZIP
	8-3-89	MINNEN CITY	Mi.	48456

This sample of handwriting is being collected for use in testing computer recognition of hand printed numbers and letters. Please print the following characters in the boxes that appear below.

0 1 2 3 4 5 6 7 8 9				
0123456789	0123456789	0123456789		
87	701	3752	80759	960941
87	701	3752	80759	960941
158	4586	32123	832656	82
158	4586	32123	832656	82
7481	80539	419219	67	904
7481	80539	419219	67	904
61738	729658	75	390	5716
61738	729658	75	390	5716
109334	40	625	4234	46002
109334	40	625	4234	46002

g y x l a k p d e b t s i r u m w f q j e n h o c v

9 Y X L A K P D E B T S I R U M W F Q J E N H O C V

Z X S B N G E C M Y W Q T K F L U O H P I R V D J A

Z X S B N G E C M Y W Q T K F L U O H P I R V D J A

Please print the following text in the box below:

We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

Figure 1: Exemple formulari NIST (4)

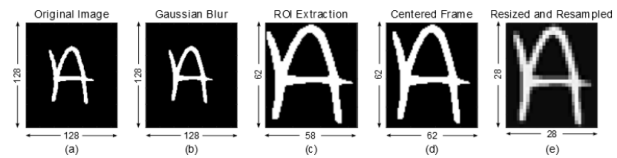


Figure 2: Transformació d'una imatge NIST a EMNIST (3)

també conté els 814,255 caràcters resultants d'aplicar segmentació de text sobre el formulari, en forma d'imatges png de dimensions 128x128 píxels, etiquetades de "0" - "9", de "A" - "Z" i de "a" - "z" en notació hexadecimal. Les imatges segmentades s'han ordenat i etiquetat en diferents organitzacions de carpetes:

- Per autor: Els 3669 formularis s'han omplert per persones diferents. Els caràcters segmentats resultants de cada formulari hi s'han agrupat per persona. Aquesta organització no és útil per OCR, sinó per diferenciar tipus d'escriptura segons individus.
- Per tipus de camp: Les imatges de caràcters s'han dividit pels diferents blocs del formulari: dígit, minúscules, majúscules i text. Seguidament s'han separat per classe de caràcter. Aquest tipus d'organització pot ser molt útil si volem fer models que només

continguin un tipus de camp. Per exemple, hi ha tasques, tals com reconeixement de caràcters de formularis, on es demana que s'escrigui només en majúscules, o potser només es volen reconèixer dígit. Entrenar un model amb un nombre de classes possibles reduït serà molt més fàcil que entrenar un model amb 62 classes. A més a més, hi ha classes de caràcters similars de camps diferents ("0", "o", "O"; "I", "l", "1"), que al redimensionar les dimensions de la imatge costaran molt d'identificar. Els diferents camps són fàcils d'identificar en una frase o paraula (majúscules només a l'inici de noms personals o frases, no barrejar números i lletres en una paraula...), però si es barregen les 62 classes a la vegada i s'intenten reconèixer, la dificultat del problema augmentarà, cosa que s'ha de tenir en compte. Aquest també és un dels motius per al que l'ús de RNN per analitzar línies senceres de text funcionen molt bé, perquè aprenen el llenguatge de manera natural, resolent automàticament el problema dels diferents tipus de camps.

- Per classe: Les imatges de caràcters estan agrupats en les 62 carpetes de les diferents classes. Amb aquesta organització, tenim el problema de no poder diferenciar entre tipus de camps.
- Per classes unides: Com que amb l'organització per classe hi ha classes massa similars entre elles, s'ha creat una nova organització ajuntant les classes entre minúscules i majúscules que s'han considerat més similars (C, I, J, K, L, M, O, P, S, U, V, W, X, Y, Z), fent un total de 47 classes diferents.

References and Notes

1. A. Chaudhuri, K. Mandaviya, P. Badelia, S. K. Ghosh, *Optical Character Recognition Systems for Different Languages with Soft Computing* (Springer, 2017), pp. 9–41.
2. J. Martínek, L. Lenc, P. Král, *Neural Computing and Applications* **32**, 17209 (2020).
3. G. Cohen, S. Afshar, J. Tapson, A. van Schaik, *CoRR* **abs/1702.05373** (2017).
4. C. I. Watson, C. L. Wilson, *Fingerprint Database, National Institute of Standards and Technology* **17**, 5 (1992).
5. Codi desenvolupat: [github/cesc1/TFG](https://github.com/cesc1/TFG)