Preprocesări de grafici vectoriale în recunoașterea scrisului de mână

Liviu-Ștefan Neacșu-Miclea[a]

[a]*Universitatea Babeș-Bolyai, Str. Mihail Kogălniceanu, nr. 1, Cluj-Napoca, 400084, Cluj, România*

Abstract

Preprocesarea datelor este un pas important în problemele de inteligență artificială. Acest articol oferă o analiză asupra câtorva variante de vectorizare a imaginilor, ale căror rezultate ar putea servi ca intrări în rețele neuronale de recunoaștere a scrisului de mână.

ACM I.2.6 Learning,
ACM I.4.0 IMAGE PROCESSING AND COMPUTER VISION General,
AMS 97M80 Arts, music, language, architecture (aspects of mathematics education),
AMS 14R99 Affine geometry

Keywords: HWR, LSTM, Potrace

1. Introducere

Handwriting recognition (HWR) este o problemă pregnantă în activitățile moderne, susținută în principal de orientarea spre automatizare și digitalizarea documentelor fizice. Având aplicații în diverse mediile oficiale, instituționale, publice sau private, tehnicile de HWR necesită o precizie extrem de ridicată, pentru ca procesarea să nu altereze informații importante, care ulterior ar trebui validate de un agent uman. Deși există numeroase abordări impresionante ale problemei, acestea nu sunt perfecte, prin urmare rămâne loc de îmbunătățiri în domeniu.

Lucrarea de față se orientează asupra aspectului de transformare a datelor astfel încât să reflecte caracteristicile esențiale în procesul de recunoaștere. Ne vom folosi de efectul procesului de vectorizare a imaginilor în a reduce dimensiunea setului de date, convertind o secvență de pixeli în secvențe de curbe Bézier de dimeniuni semnificativ mai reduse, permițând în același timp segmentarea la nivel de cuvânt și procesarea secvențială a datelor.

2. Related work

2.1. Modele folosite

HWR, ca suproblemă OCR, are o bogată istorie de abordări. Există atât recunoaștere online, care se folosește de input-ul în timp real al unui dispozitiv extern, cum ar fi o tabletă grafică, cât și recunpaștere offline, în care textul de mână este scanat, de obicei sub formă de imagine, și procesat ulterior. În acest sens, se folosesc diverși algoritmi bazați pe image feature extraction, pentru care rețelele tip CNN sunt o soluție foarte comună. Un model CNN se folosește de convoluții pentru a detecta features de complexitate graduală (mai întâi linii și curbe, forme, apoi context). Aceste modele produc rezultate excepționale în recunoașterea caracterelor izolate (cum ar fi cele din setul de date EMNIST), cu precizii de peste 99%. Cu toate acestea, principala problemă într-o astfel de metodă este însăși segmentarea caracterelor. Scrisul de mână este de obicei continuu, iar în funcție de scriitor caracterele pot fi înclinate, sau chiar rândul textului poate devia de la direcția orizontală cu cât se apropie de marginea dreaptă a paginii. De obicei, se folosesc procesări precum deskew sau slant correction pentru a corecta aceste dereglări, de obicei detectând unghiul de înclinare al textului folosind liniile Hough. Apoi, se încearcă identificarea literelor aplicând euristici asupra curbei imaginare care învelește marginea superioară a cuvântului. Alte metode folosesc un sliding window pentru a parcurge o linie de text. În acest context, CNN devine ineficient și poate fi înlocuit cu modele convoluționale mai performante ca FCNN, care își ating până la urmă limita, lăsând loc arhitecturilor de tip RNN să-și dovedească aplicabilitatea în acest domeniu. Uneori, combinarea modelelor CNN și LSTM a condus la rezultate remarcabile. Mai recent, au devenit populare modelele transformer, aplicate cu success și în recunoașterea vocală, problemă care este tratată asemănător cu recunoașterea de text, din cauza multiplelor puncte de convergență.

2.2. Metrici uzuale

Măsura calității unui text prezis se poate decide folosind distanța Levenshtein, care calculează numărul minim de modificări necesare pentru a transforma un text în altul. Character Error Rate (CER) măsoară procentul de caractere care au fost prezise incorect de către model. Analogul său, Word Error Rate (WER), consideră cuvântul ca unitate atomică de decizie. Însă, poate cea mai interesantă măsură este cea produsă de Connectionist Temporal Classification (CTC), care folosește mai multe clase decât au fost declarate inițial ca output, alegand rezultatul in urma unei decizii probabilistice asupra tuturor acestor clase. CTC foloseste un RNN pentru a calcula probabilitatile la fiecare pas/moment de timp si emite o tabla de

Figure 1: Etapele de transformare ale unui input raster intr-unul vectorial

incredere care include fiecare clasa (ex. litera/sunet) si probabilitatea corespunsatoare. Măsura CTC trebuie să se apropie de 0 pentru a ne asigura că o predicție este corectă.

2.3. Seturi de date

Setul de date EMNIST cuprinde peste 60000 de imagini 28x28px cu caractere individuale. Subseturi ale sale iau în considerare uniformizarea claselor (litere mici și mari să reprezinte aceeași clasă) sau echitabilitatea (balansarea) datelor.

Setul de date IAM conține fragmente de text întregi scrise de numeroși autori. Este folosit atât pentru recunoașterea textului, cât și pentru detectarea amprentei caligrafice a autorului.

2.4. Metode extrase din lucrări existente

Metoda descrisa in [2] folosește o regresie non-lineară pentru a segmenta curbele din imagine, care ar putea fi pus în dificultate în condițiile în care grosimea liniei este mărită. Sunt folosite concepte din fizică și psihologice combinate cu analize statistice pentru recunoașterea caracterelor. In [6] a fost folosit un model LSTM multidimensional cu clasificator CTC, dar pentru dataset cu scripturi atât latine, cât și arabice, raportând precizie de peste 90TextCaps [11] foloseste FCNN + capsule networks pentru a identifica caracterele din EMNIST cu 95acuratete. Metoda propusa de autori este utila pentru augmentarea setului de date si nu este destinata recunoasterii, modelul nostru este aplicabil in acest context si are potential de a oferi rezultate promitatoare. In [3.] s-a folosit o structura CNN-RNN 50-98% rata de identificare corecta a cuvintelor pe setul de date, folosind CTC ca metrica de evaluare. Autorii discuta si influenta ratei train-test in determinarea preciziei finale. De asemenea, sunt abordate metode de segmentare a textului cu OpenCV, care in contextul lucrarii curente sunt echivalentul vectorizarii. Lucrarea [13] aplica GRU pe IAM cu 95% acuratete. Functia de loss folosita este cross-entropy. Modelul descris are si aplicatii generative.

3. Metoda propusă

Ideea de bază a proiectului constă în convertirea imagiinlor din setul de date IAM_words din formatul raster intr-un format vectorized. A fost ales algoritmul Potrace pentru rezultatele sale satisfăcătoare oferite într-un timp decent. O implementare proprie se regăsește în cadrul proiectului, în cazul necesității unor modificări ulterioare.

Fie $\mathscr{C}$ multimea spațiului de culori si $\mathscr{G} : \mathscr{M}_{h,w}(\mathscr{C}) \to \mathscr{M}_{h,w}([0,1])$ o funcție de conversie a imaginii în tonuri de gri (grayscale), și $\gamma \in [0,1]$ un factor de threshold. Fie $P = (p_{i,j})_{1\le i\le h, 1\le j\le w} \in \mathscr{M}_{h,w}(\mathscr{C})$ un input din setul de date. Se construiește imaginea binară $P' = (p'_{i,j})_{1\le i\le h, 1\le j\le w}$, unde $p'_{i,j} = \begin{cases} 0 & , \mathscr{G}(P)_{ij} < \gamma \\ 1 & , \text{altfel} \end{cases}$ și se aplică algoritmul *Potrace* : $\mathscr{M}_{h,w}([0,1]) \to (\mathbb{R}^7)^*$, unde $X^* = \{(x_1, ..., x_k) | k \ge 0, x_i \in X \forall i = \overline{(1,k)}\}$ este multimea listelor ordonate cu elemente din X . Un element $b = (x_1, y_1, x_2, y_2, x_3, y_3, \alpha) \in \mathbb{R}^7$ descrie o curbă Bezier de gradul 3 ce are ca parametri punctele $P_1 = (x_1, y_1), P_2 = (x_2, y_2), P_3 = (x_3, y_3)$, și coeficientul de curbură α ,al cărei poligon de control este determinat de segmentele P_1P_2, P_1P_3, P_2P_3 și paralela la P_2P_3 care taie P_1P_2 în segmente de proporții $(1-\alpha), \alpha$.

De alegerea funcției $\mathscr{G}$ si a factorului γ depinde calitatea outputului oferit de *Potrace*. Blurarea imaginii poate fi de asemenea un factor decisiv pentru mai buna identificare a curbelor.

3.1. Antrenarea modelului

Avantajul preprocesării anterioare este acela că inputul nu mai este limitat de o dimensiune fixă a imaginii, și că există posibilitatea unei procesări secvențiale a acestuia. Astfel, în cadrul unui model de rețea neuronală, putem înlocui tradiționalul CNN cu o celulă LSTM imediat atașată inputului. Ulterior, două layere de activare reduc dimensiunile feature-urilor la 64, care sunt trecute prin două celule LSTM bidirecționale. Rezultatul este adus la o matrice care are, pe linii, numărul lungimea maximă a unei secvențe care poate fi recunoscută dintr-un input, respectiv numărul maxim de caractere admise, plus încă un blank character, folosit ulterior de către CTC.

3.2. Clasificarea

Măsura folosită pentru loss este Connectionist Temporal Classification (CTC), specializată pentru output-ul RNN. Acuratețea modelului se decide pe baza inputurilor pentru care ctc_loss=0. (care prezic cu exactitate outputul)

3.3. Posibile metode de preprocesare

3.3.1. Identificarea curbelor prin rețele ANN

Inițial, este de menționat că s-a încercat tratarea vectorizării ca o problemă de sine stătătoare. În acest sens, am încercat o rețea convoluțională specializată în detectarea anumitor tipuri de curbe în fiecare chunk de 32x32px al unei imagini. Avantajul ideii era aceea că tracing-ul ținea cont și de grosimea liniilor, ceea ce producea outputuri similare cu cele oferite de dispozitivele fizice folosite în online OCR (asemanator [2]), și nu o simplă înfășurătoare geometrică a zonei de interes.

3.3.2. Vectorizarea Potrace

Reteaua a fost antrenata pe un set mic de 10 date de intrare. Acuratetea tinde sa convearga mai lent decat in cazul modelului CNN-BiLSTM, in schimbul posibilitatii de a propcesa inputuri de lungimi variate.

In ambele cazuri, variatia acuratetei depinde de evolutia lossului, care trebuie sa fie sub un anumic prag pentru a asigura o crestere a ratei de invatare.

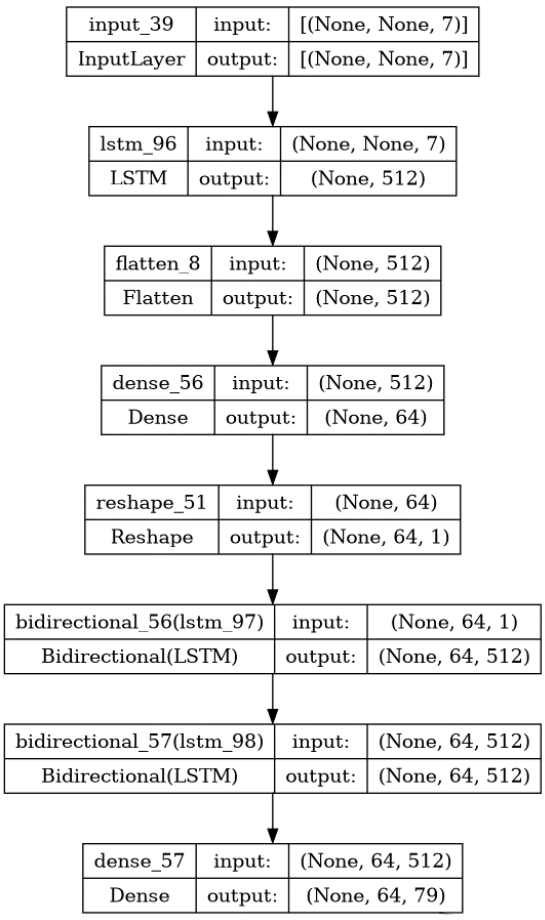

Figure 2: Arhitectura rețelei LSTM

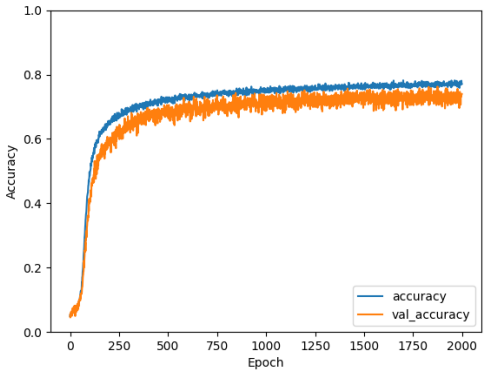

Figure 3: Metricile de acuratețel pentru rețeaua ANN

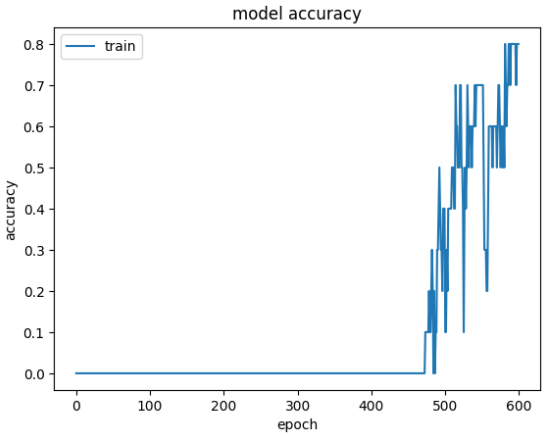

Figure 4: Evoluția modelului LSTM

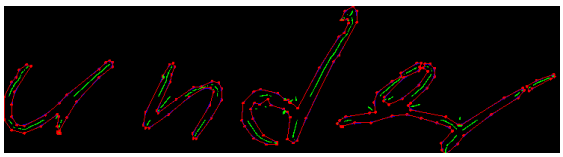

Figure 5: Retrasarea (cu verde)

3.3.3. *De-poligonare*

S-a observat în secțiunile anterioare că Potrace generează un contur al cuvântului, ceea ce este oarecum neintuitiv comparativ cu trasarea naturală a textului pe o bucată de hârtie. O posibilă îmbunătățire ar putea fi retrasarea scrisului folosind un stilou imaginar care se poate deplasa doar în interiorul conturului cuvântului, fără a-l depăși, și păstrând tot timpul o distanță echilibrată de laturile poligonului.

Pentru aceasta, vom păstra din fiecare curbă Bézier doar poligonul de control și vom unifica segmentele continue (dând practic Potrace cu un pas în urmă). Se obține o mulțime de segmente în plan, originea acestora inițială fiind neimportantă, care reprezintă forma geometrică a cuvântului. Aceasta poate fi neconexă și poate conține găuri. Algoritmul propus consideră fiecare punct din interiorul formei. Acestea se pot calcula ușor folosind algoritmul de testare al apartenenței unui punct la un poligon folosind raze orizontale. Daca segmentul determinat de punctele $(x, 0)$ și (x, y) intersectează un număr impar de segmente din plan, atunci punctul (x, y) se află în interiorul formei. Pentru fiecare punct (x, y), fie $s_k, k = \overline{1, n}$ segmentele care se intersectează o vecinătate circulară de rază r a acestuia. Definim neutralitatea punctului $N(x, y)$ o măsură care specifică dacă un punct este echitabil depărtat de segmentele care intersectează vecinătatea sa. Formula aleasă pentru calculul acestei metrici este $N(x, y) = \sum_i \frac{|d_i - \overline{d_i}|}{n}$, unde $d_i = dist((x, y), s_i)$ este distanța de la punctul (x, y) la segmentul s_i. Prin alegerea unui prag ϵ apropiat de 0, punctele pentru care $N(x, y) < \epsilon$ pot fi considerate neutre. Punctele neutre sunt o referință pentru reconstruirea curbei cuvântului așa cum ar fi el scris in realitate. Rezultatul acestei metode poate fi folosit chiar împreună cu algoritmii de recunoaștere online.

4. Rezultate

În cazul modelului ANN de detectare a curbelor Bezier, desi a fost antrenat cu precizie medie, performanțele acestei metode au fost mult mai scăzute in realitate, din cauza lipsei unui set de date specializat pe această problemă. Modelul a fost antrenat pe inputuri generate aleator într-un mediu controlat și manifestă fenomenul de overfitting, fiind inutilizabil chiar și pe inputuri simple.

Același definit de date adaptat pe problemă îngrădește potențialul de testare a capacităților preprocesării cu Potrace. Pe selecția extrasă din setul de date IAM, loss-ul modelului converge foarte greu (sau chiar deloc). (Cel mai probabil nu am aplicat LSTM-ul corect). Pentru 10 date de antrenament, s-a ajuns la precizie 80% abia după 600 de epoci. O eroare umană este de asemenea cauza unei estimări imprecise, motivul fiind segmentarea incorectă a unor cuvinte, producând o linie bazală orizontală care unește cuvinte diferite, contrazicând motivația fundamentală a acestei lucrări (segmentarea pe cuvânt și procesarea secvențială).

Este de așteptat ca depoligonarea să producă rezultate mai bune, întrucât se analizeaza cca. 50% din datele generate de Potrace, fără a pierde informație utilă.

5. Concluzii

Vectorizarea imaginilor care conțin text deschide drumul către abordări de tip secvențial în procesul de recunoaștere. Textul este prin natura sa un element cu o structură geometrică predictibilă și indefinit repetabilă, lucru pe care rețelele au dificultăți în a-l valorifica. În ciuda acțiunilor neintenționate de auto-sabotare ale autorului, potențialul ideii nu trebuie neglijat. Metoda descrisă anterior este fiabilă într-un mediu curat și controlat. Existența unor elemente adiționale în imagine (de exemplu, liniile unui caiet) pot fi interceptate de metodele de vectorizare, invalidându-le scopul. Asemenea artefacte ar trebui filtrate prin metode specializate. În viitor, se poate pune accentul pe rafinarea metodei descrise, îmbunătățirea setului de date, crearea unei arhitecturi neuronale adecvate și optimizarea performanțelor algoritmilor de preprocesare.

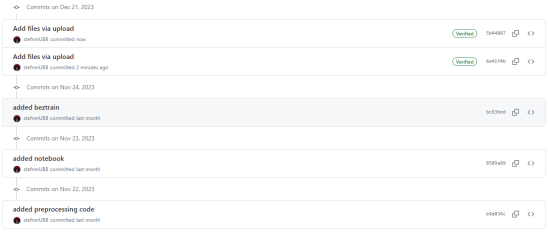

Figure 6:

References

[1] C. C. Tappert, C. Y. Suen and T. Wakahara, "The state of the art in online handwriting recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 8, pp. 787-808, Aug. 1990, doi: 10.1109/34.57669.

[2] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63-84, Jan. 2000, doi: 10.1109/34.824821.

[3] L. Xu, A. Krzyzak and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, no. 3, pp. 418-435, May-June 1992, doi: 10.1109/21.155943

[4] C. Bahlmann, B. Haasdonk and H. Burkhardt, "Online handwriting recognition with support vector machines - a kernel approach," Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, Niagra-on-the-Lake, ON, Canada, 2002, pp. 49-54, doi: 10.1109/IWFHR.2002.1030883.

[5] Jianying Hu, M. K. Brown and W. Turin, "HMM based online handwriting recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 1039-1045, Oct. 1996, doi: 10.1109/34.541414.

[6] A. Graves, J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks," in Advances in Neural Information Processing Systems, 2008.

[7] A. W. Senior and A. J. Robinson, "An off-line cursive handwriting recognition system," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 309-321, March 1998, doi: 10.1109/34.667887.

[8] Sun, J., et al. "Image Vectorization Using Optimized Gradient Meshes," in ACM Trans. Graph., vol. 26, no. 3, pp. 11–es, 2007.

[9] Lejun Shao. Hao Zhou. "Curve Fitting with Bézier Cubics," in Graphical Models and Image Processing, vol. 58, no. 3, pp. 223-232, 1996.

[10] Yuxing Tan, Hongge Yao. "Deep Capsule Network Handwritten Digit Recognition," in International Journal of Advanced Network, Monitoring and Controls, vol. 5, no. 4, pp. 1–8, 2020.

[11] V. Jayasundara, S. Jayasekara, H. Jayasekara, J. Rajasegaran, S. Seneviratne and R. Rodrigo, "TextCaps: Handwritten Character Recognition With Very Small Datasets," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2019, pp. 254-262, doi: 10.1109/WACV.2019.00033.

[12] R. Sumathy, S. N. Swami, T. P. Kumar, V. L. Narasimha and B. Premalatha, "Handwriting Text Recognition using CNN and RNN," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 766-771, doi: 10.1109/ICAAIC56838.2023.10140449.

[13] Dr. S. Kanmani, B. Sujitha, K. Subalakshmi, S. Umamaheswari, Karimreddy Punya Sai Teja Reddy. "Off-Line and Online Handwritten Character Recognition Using RNN-GRU Algorithm," in International Journal for Research in Applied Science Engineering Technology (IJRASET), 2023, pp. 2518-2526, doi:10.22214/ijraset.2023.50184.

[14] Marwa Dhiaf., et al. "MSdocTr-Lite: A lite transformer for full page multi-script handwriting recognition," in Pattern Recognition Letters, vol. 169, pp. 28-34, 2023.

[15] T. Bluche, J. Louradour and R. Messina, "Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 1050-1055, doi: 10.1109/ICDAR.2017.174.