

Prepoznavanje japanskog pisma - Hiragane

Bojan Batalo, Dušan Radisavljević Profesor: Đorđe Obradović Asistent: Miroslav Kondić



Motivacija

Prepoznavanje karaktera nalazi široku primenu u današnje vreme kao pomoć gluvima, obradi dokumenata, prepoznavanju stranog jezika i mnogim drugim problemima. Iako je problem široko istražen za latinski alfabet, problematika prepoznavanja hiragane je gotovo neistražena.

Cili:

Prepoznati digitalno unete karaktere uprkos šumu.

Dataset i obrada

S obzirom da problem prepoznavanja hiragane u prošlosti nije toliko rešavan, očekivano je bilo da će biti mala količina datasetova dostupnih na internetu. Pretražujući mrežu, naišli smo na dva dataseta, za koje nismo smatrali da su pogodni za problem koji smo želeli rešiti. Usled toga odlučili smo načiniti dataset za sopstvene potrebe, koji bi se dalje mogao koristiti u rešavanju problematike. Sa 9 slika iskucane na 3 različita fonta koje obuhvataju sva slova hiragane ručno su izvučeni svi karakteri i načinjena hdf5 datoteka. Izvlačenje se postiže pozivom skripte make_images.py, dok se kreiranje datoteke postiže pozivom prep_dataset.py.

dataset.py skripta sadrži poziv za učitavanje dataseta u memoriju na način na koji se MNIST dataset čuva u memoriji. Pozivom metode require_new(train,test) kreiraju se liste train i test podataka u memoriji, na osnovu navedenih brojeva, kao i lista nizova labela koja odgovara listi podataka. Moguće je pozvati i request_new_RNN(test,train) koja učitava listu labela, umesto liste nizova labela.

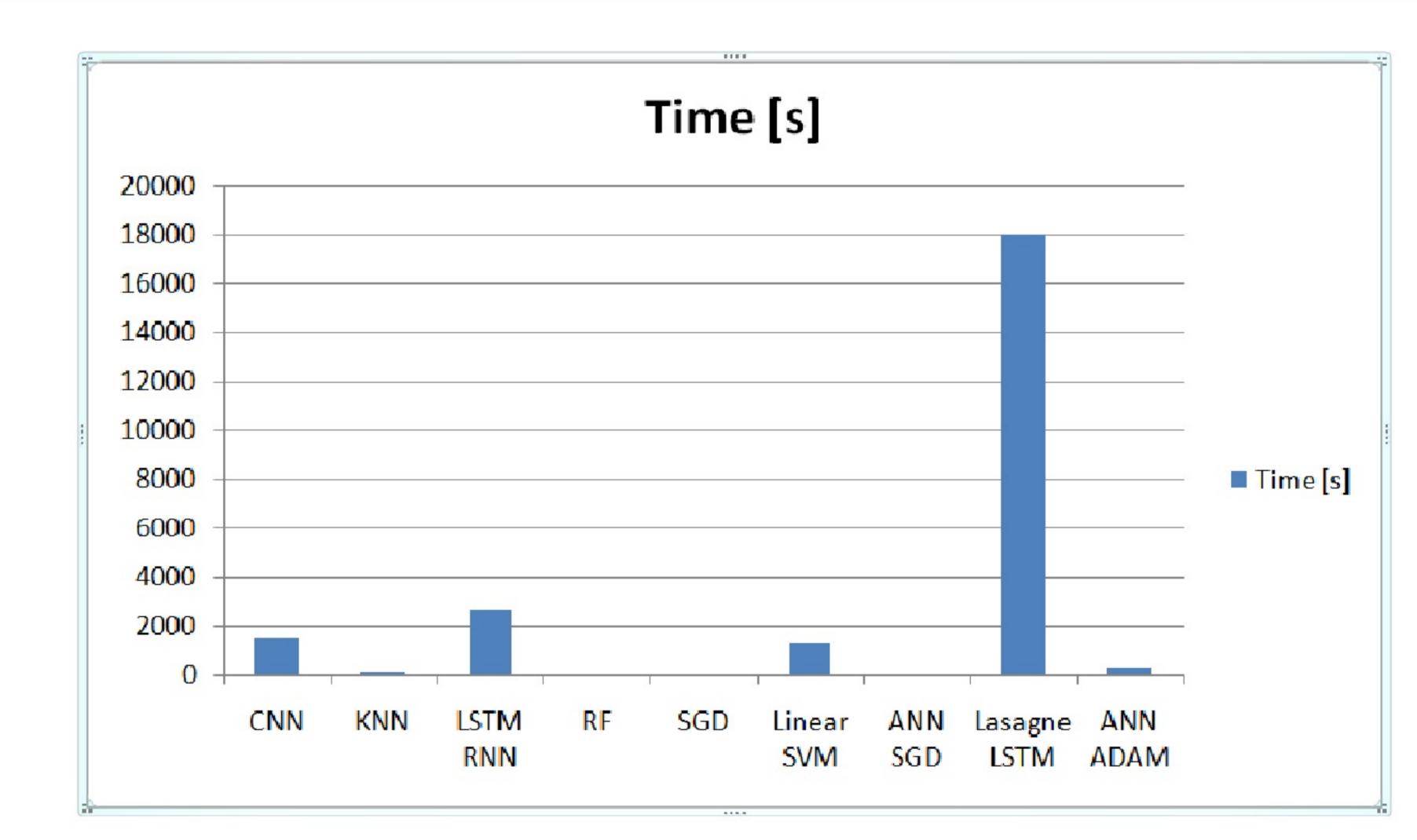
Proces rada

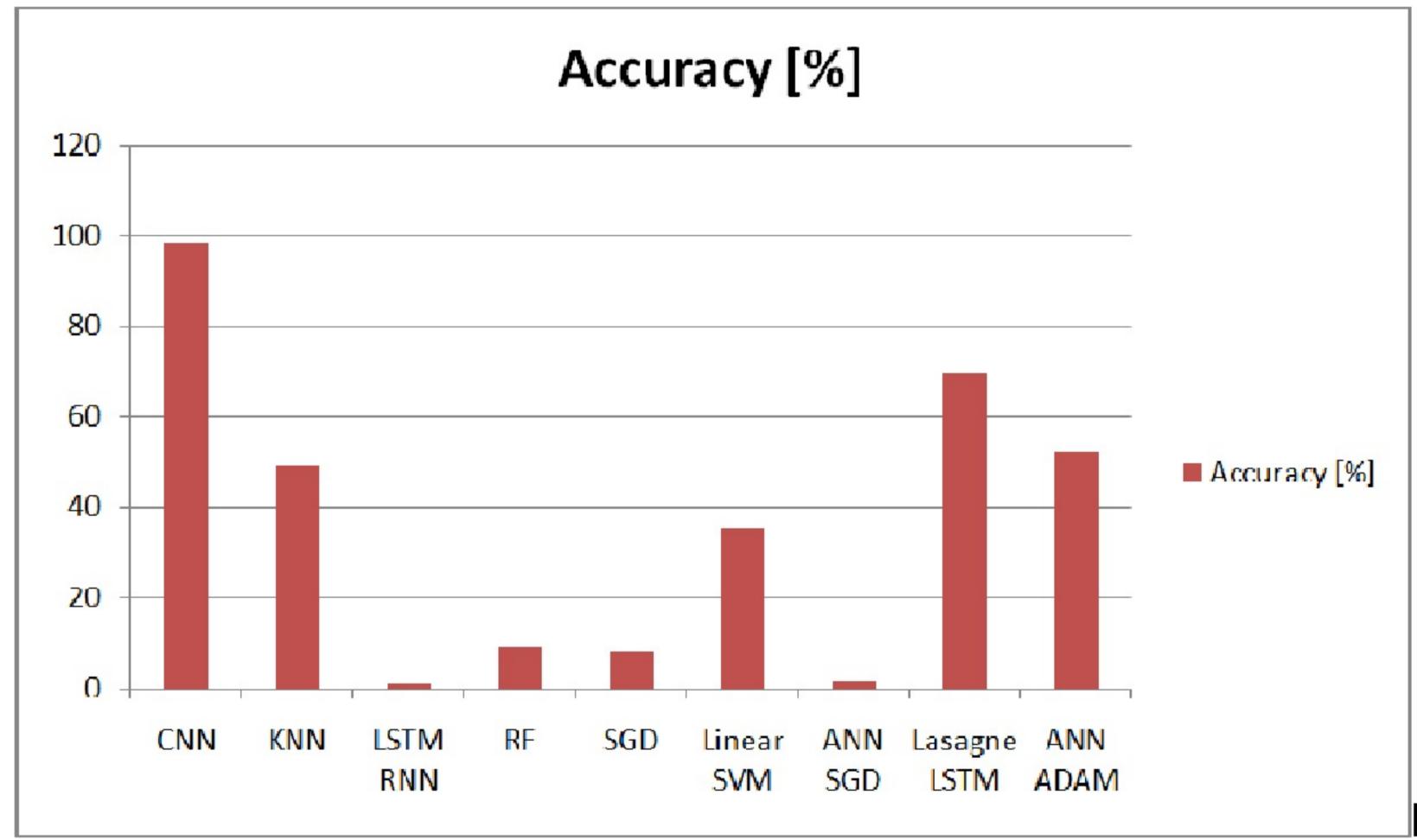
Proces izdvajanja regiona: Slika koja se testira pretvara se putem segmentacije i morfoloških operacija u prikladnu binarnu verziju, sa crnom pozadinom i belim istaknutim delovima. Beli delovi se dalje analiziraju, pri cemu se izdvajaju početni regioni. Kako nas zanimaju samo regioni određene veličine, izračunava se maksimalna širina/visina regiona (pretpostavka je da nema regiona većih od samih karaktera), pa se izbacuju oni koji su mnogo manji od tog maksimuma. Pošto su karakteri Hiragane takvi da se mnogi od njih pišu iz više poteza, odnosno njihovi regioni su razdvojeni, potrebno je spojiti te regione kako bi se izdvojio sam karakter. Uvideli smo četiri nacina na koji su regioni razdvojeni, a to su: regioni jedan iznad drugog, regioni jedan pored drugog, regioni kojima se bounding box preklapa, i regioni kojima se u gornjem desnom uglu nalaze drugi delovi regiona. Dalje se postavlja pitanje kojim redosledom primeniti ove operacije, tako da se regioni ne preklope na pogrešan način, za šta postoji jako velika šansa. Eksperimentalno ustanovljen redosled: preklapanje x2 ->goredole, preklapanje, levo-desno, preklapanje, gore-desno, preklapanje, goredesno, prekapanje. Izdvajaju se granice regiona, koje se dalje koriste kako bi se sa originalne grayscale slike izvukli karakteri (prediktori rade na grayscale slikama), te se prosleđuju neuronskoj mreži.

Od implementacija neuronske mreže unutar fajla hnn.py nalaze se Konvoluciona neuronska mreža, k najbližih suseda i dvoslojna veštačka neuronska mreža sa softmax aktivacijom. Uz to priložen je fajl RNN.py koji sadrži implementaciju višeslojne rekurentne neuronske mreže, mada je njeno obučavanje, usled nedostatka memorije, limitirano. Unutar Neural_Network_Zoo ipython notebook fajla nalaze se još implementacije LSTM rekurentne neuronske mreže, Random Forest, Stochastic Gradient Descent, linearni Support Vector Machine kao i veštačka neuronska mreža sa SGD aktivacijom. Svaka od ovih neuronskih mreža je trenirana i sadrži informacije o trajanju obučavanja i preciznosti postignutoj na test skupovima u komentarima ćelije.

Uz njih se nalazi i VGG16 neuronska mreža sa učitanim težinama, koja nije praktično testirana.

Procena i rezultati





Kao što se može primetiti sa grafika, najbolji rezultati su ostvareni upotrebom konvolucione neuronske mreže, gde se na test primerima ostvaruje preciznost od preko 98%.

Predlog za dalje usavršvanje bi bila upotreba rekurentne neuronske mreže, ali sa većim memorijskim kapacitetom i širim obučavajućim skupom.