

Detecția defectelor în industria porțelanului folosind rețele neuronale adânci

Autori: Onița Daniela Marcela, Golban Igor

Profesor coordonator: Lect. univ. dr. Bîrluțiu Adriana

Instituție: Universitatea “1 Decembrie 1918” din Alba Iulia

Abstract

This application was created to help companies specialized in porcelain industry to detect defects on the products. At this moment, in most of these companies defect detection is performed manually by employees. It is needed an intelligent system for product monitoring and this application is an essential step to achieve it. In order to implement this intelligent system we proceeded as follows. First, it was defined a convolutional neural network (CNN) in Python, using Keras and Theano libraries and deep learning techniques. Second, OpenCV library is used to read new images (which must be predicted). Third, CNN will analyze and will predict if the input images have defect or not have. In the future, with a specific implementation, the application can be used in other areas where product quality is essential.

1. Introducere

În această lucrare este descrisă crearea unei aplicații care are rolul de a detecta defectele unor farfurii de porțelan create de companiile specializate în industria porțelanului.

Procesul de fabricație a porțelanului constă din următoarele etape:

- 1) Prepararea masei ceramice;
- 2) Atomizarea pentru obținerea pulberii atomizate;
- 3) Fasonarea sau presarea obiectului;
- 4) Biscuitarea (Arderea I);
- 5) Glazurare;
- 6) Arderea II;
- 7) Sortarea finală.

Verificarea calității și înlăturarea produselor defecte se face după biscuitare, glazurare și la sortarea finală. Cerințele de calitate ale clienților au devenit foarte ridicate, companiile active în industria porțelanului fiind obligate să livreze numai produse de calitate întâi. Concomitent, s-au diversificat criteriile de inspecție a produselor, a crescut numărul de produse inspectate și complexitatea sarcinilor de control. Defectele produselor ceramice se datorează unor factori și parametri ai procesului tehnologic. Astfel ele pot fi clasificate în mai multe categorii precum: asimetrii, curburi, margini deformate, culoare degradată, scurgere de glazură, urme de retușare, exfolieri, fisuri, crăpături, știrbituri, zgârieturi, etc.

În acest moment, în majoritatea companiilor din industria porțelanului, verificarea aspectului produselor și identificarea defectelor se realizează vizual și prin palpare, manual, de către angajați. Acest lucru nu este automatizat, deoarece se dorește ca cerințele de control a calității să afecteze cât mai puțin costul și timpii de producție.

Companiile active în industria porțelanului au nevoie de un sistem inteligent implementat în fluxul tehnologic care să fie conectat și să comunice cu roboții industriali. Sistemul trebuie să monitorizeze și să controleze defectele produselor din porțelan, în vederea eliminării acestor produse defecte după fiecare fază de fabricație.

2. Tehnologii folosite

2.1 Rețele neuronale adânci (*Deep learning*)

Deep learning este considerată o ramură a “machine learning”, care are ca obiectiv apropierea dintre machine learning și inteligența artificială (*Artificial Intelligence*), prin învățarea computerelor a unor lucruri de care în trecut doar oamenii erau capabili (recunoașterea imaginilor, a vocii, etc.).

Deep learning este considerată a fi un set de tehnici de învățare în rețele neuronale artificiale, care utilizează multiple straturi/nivele (*layers*) pentru a rezolva diferite probleme de recunoaștere.

Figura 1 ilustrează comparația dintre algoritmi bazați pe tehnici de deep learning și alți algoritmi de învățare. Din grafic este evident faptul că în cazul deep learning cu cât cantitatea datelor crește, cu atât crește și performanța algoritmilor, în timp ce în cazul altor algoritmi de învățare nu se întâmplă acest lucru.

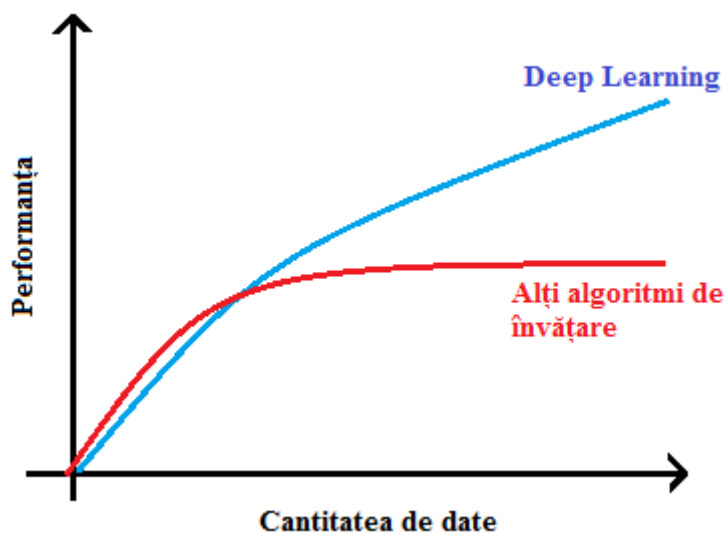


Figura 1: Comparație deep learning – alți algoritmi de învățare [5]

Rețelele neuronale artificiale sunt definite ca sisteme neprogramate de procesare adaptivă a informației, procesarea informației fiind neexplicită. Acestea sunt organizate în straturi/nivele, care sunt definite ca fiind noduri interconectate ce conțin funcția de activare, funcție care este componentă în structura neuronului artificial. Modele sunt reprezentate printr-un layer de intrare (*input layer*) care comunică cu unul sau mai multe “layer-e ascunse” (*hidden layers*), care la rândul lor sunt legate de layer-ul de ieșire (*output layer*).

Reprezentarea nivelelor poate fi vizualizată grafic în Figura 2. Nivele ascunse sunt numite ascunse datorită unităților care nu sunt conectate direct cu mediul înconjurător. [2]

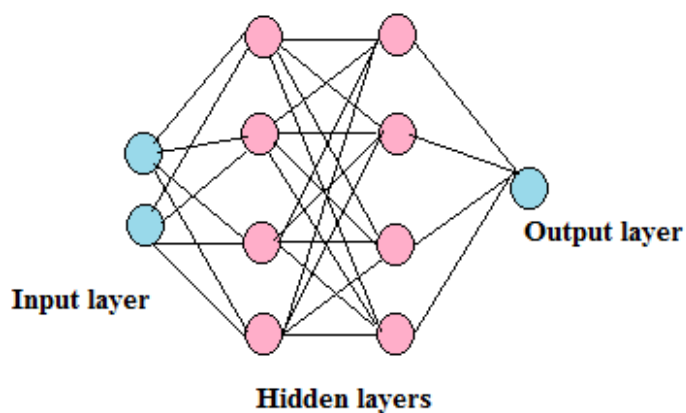


Figura 2: Reprezentarea nivelelor rețelei neuronale [1]

2.1 Învățare supervizată

În cadrul învățării supervizate vom primi un set de perechi (x, y) , cu scopul de a găsi o funcție f , astfel încât funcția să aproximeze cât mai bine exemplele date.

Învățarea supervizată este echivalentă cu optimizarea unei funcții de eroare care măsoară diferența dintre răspunsurile pe care ar trebui să le producă rețeaua și cele pe care le produce efectiv.

Pentru rețeaua neuronală tratată la nivelul lucrării se aplică învățarea supervizată. Setul de antrenare în cazul acestui tip de învățare conține atât date de intrare, cât și răspunsurile corecte aferente acestora, deoarece există atât farfurii corecte, cât și farfurii cu defecte.

2.2 Rețele neuronale convoluționale (*Convolutional neural networks*)

Rețelele neuronale convoluționale fac parte din categoria rețelelor neuronale feed-forward în care modelul de conectivitate între neuroni este inspirat de organizarea cortexului vizual al animalelor. Au aplicabilitate în probleme de recunoaștere a imaginilor, respective probleme de clasificare.

La fel ca și în cazul rețelelor neuronale simple, arhitectura rețelelor neuronale convoluționale este formată din:

- Convolutional layer
- Pooling layer
- Fully-connected layer

Rețelele neuronale convoluționale au straturi de convoluție, prin care se definește rețeaua, straturi care se aplică peste stratul de intrare (input layer), pentru a determina stratul de ieșire (output layer). Fiecare strat va aplica diferite filtre asupra datelor, care puse împreună vor fi capabile să facă clasificarea. Filtrele aplicate asupra datelor pot fi să determine marginile pixelilor, fie să detecteze forme simple, fie caracteristici de nivel înalt, pentru ca în final, ultimul strat să poată face clasificare folosind acele caracteristici de nivel înalt.

3. Rezultate experimentale

3.1 Limbajului și librăriile utilizate

Limbajul folosit pentru implementarea problemei tratate la nivelul acestei lucrări este Python [6]. Principalele biblioteci utilizate pentru crearea aplicației sunt: Keras [7], Theano [8] și OpenCV [9].

Python este un limbaj de programare popular, care oferă atât posibilitatea programării

structurate, cât și a celei orientate pe obiect și include și elemente din paradigma funcțională. Este un limbaj de scripting, deci este interpretat, nu compilat, astfel se economisește mult timp în procesul de dezvoltare și debugging.

Keras este o bibliotecă pentru rețele neuronale de nivel înalt API, scrisă în Python și capabilă să ruleze fie cu biblioteca TensorFlow, fie cu Theano. În acest caz Keras rulează împreună cu Theano.

Theano este o bibliotecă Python care permite definirea, optimizarea și evaluarea expresiilor matematice implicand eficiența vectorilor multi-dimensionali.

Biblioteca **OpenCV** conține peste 2500 de algoritmi, care pot fi utilizați în vederea artificială, de exemplu pentru a detecta și recunoaște fețe, pentru a identifica obiecte, clasifica acțiunile umane în videoclipuri, mișcările camerei, etc.

3.2 Setul de date folosit

Setul de date folosit pentru aplicația descrisă la nivelul acestei lucrări conține farfurii fabricate de o companie specializată în industria porțelanului. În imagini sunt ilustrate atât farfurii cu defecte, cât și farfurii corecte. Defectele întâlnite pe suprafața farfuriilor pot fi fie fisuri, fie deformări, pete sau defecte de textură. În Figura 3 se pot observa câteva dintre acestea:



Figura 3: Defectele întâlnite pe suprafața farfuriilor

Pentru ca imaginile să poată fi încărcate, a fost nevoie de redimensionarea imaginilor și de setarea căilor către fișierele care conțin datele de antrenare, respective cele de testare. Acestea au fost încărcate folosind mediul de programare Spyder [10], inclus în managerul de mediu de programare Anaconda [11].

3.3 Descrierea aplicației

Pentru a ajunge la varianta finală a aplicației, a fost nevoie de crearea modelului rețelei prin definirea nivelelor rețelei, prin compilarea, antrenarea și evaluarea modelului.

```
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])
```

Figura 4: Definirea straturilor rețelei și compilarea modelului creat

Ca și modificări/augumentări a datelor, a fost nevoie de rescalarea datelor de antrenare și a datelor de testare, modificări ilustrate în Figura 5.

```
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

# this is the augmentation configuration we will use f
# only rescaling
test_datagen = ImageDataGenerator(rescale=1. / 255)
```

Figura 5: Augumentarea datelor

În Figura 6 este prezentat output-ul rețelei neuronale. Se poate observa faptul că rețeaua a găsit în directoarele indicate 121 de imagini pentru antrenare, respective 40 de imagini pentru testare, acestea aparținând de 2 clase (farfurii corecte, respective farfurii cu defecte), deci este o problem de clasificare binară.

```

Found 121 images belonging to 2 classes.
Found 40 images belonging to 2 classes.
Epoch 1/30
31/31 [=====] - 270s - loss: 0.7497 - acc:
0.6504 - val_loss: 0.6383 - val_acc: 0.6944
Epoch 2/30
31/31 [=====] - 247s - loss: 0.6343 - acc:
0.6641 - val_loss: 0.5511 - val_acc: 0.7361
Epoch 3/30
31/31 [=====] - 250s - loss: 0.6335 - acc:
0.6708 - val_loss: 0.5622 - val_acc: 0.7639
Epoch 4/30
17/31 [=====>.....] - ETA: 100s - loss: 0.6072 - acc:
0.6949

```

Figura 6: Output-ul rețelei neuronale adânci

Figura 7 ilustrează acuratețea obținută în raport cu numărul de epoci. Linia albastră reprezintă acuratețea pentru datele de antrenare, iar linia albastră reprezintă acuratețea pentru datele de test. Ca și medie pentru acuratețea datelor de testare s-a obținut valoarea 0.75.

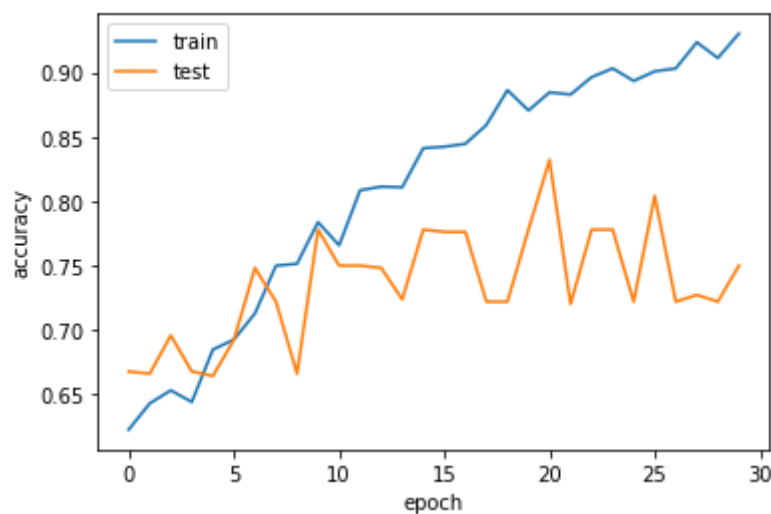


Figura 7: Acuratețea obținută în raport cu numărul de epoci (epochs)

În a doua parte a aplicației s-a utilizat biblioteca OpenCV pentru detecția defectelor de pe farfurii. Cu ajutorul acestei biblioteci se vor citi noile imagini pentru care se dorește controlul calității (se va controla dacă are sau nu defecte). Rețeaua neuronală va face analiza noilor imagini și va returna predicția dacă farfuria are sau nu defect.

4. Concluzii și direcții viitoare de dezvoltare

Această lucrare poate fi considerată un plus pentru procesul de înlocuire a forțelor de lucru umane cu mașinăriile/ roboții. La nivelul ei este descrisă o aplicație care îmbunătățește performanța

companiilor specializate în industria porțelanului, prin detectarea automată a defectelor întâlnite pe suprafața produselor fabricate.

După implementarea sistemului inteligent de detecție a defectelor de pe farfurii nu va fi afectat nici costul, nici timpul și nici calitatea produselor. În plus, considerentele economice ale companiilor active în indutria porțelanului se vor îmbunătății, deoarece odată ce banii sunt investiți în implementarea sistemului, nu se vor mai cheltui bani pentru angajații care se ocupau de controlul calității.

Pe viitor, cu o implementare adecvată, aplicația va putea fi folosită pentru orice alt tip de companie în care este nevoie de controlul calității produselor fabricate, spre exemplu o companie din industria pavajelor.

Bibliografie

- [1] Jiawei Han and Micheline Kamber, “Data mining – Concepts and Techniques”, Editura Morgan Kaufmann, 2006
- [2] Lucrare de licență Onița Daniela Marcela, “Extragerea cunoștințelor din seturi mari de date utilizând tehnici de data mining și agenți inteligenți”
- [3] Lucrare de licență Cristea Radu, “Detecția automată a anomaliilor în mamografii”
- [4] Jason Brownlee, “Deep Learning With Python”, 2017
- [5] <http://machinelearningmastery.com/what-is-deep-learning/> (11 aprilie 2017)
- [6] <https://www.python.org/>
- [7] <https://keras.io/>
- [8] <http://deeplearning.net/software/theano/>
- [9] <http://opencv.org/>
- [10] <https://anaconda.org/spyder-ide>
- [11] <https://anaconda.org/>