Czy mogę pisać CNN, SVM? - skróty na początku

CZy opisywać same technologie? Chyba ta

Ogólnie:

Autor znalazł tylko jedną pracę zajmującą się rozpoznawaniem tylko ?pełnych? odpadów plastikowych. Zazwyczaj badacze skupiali się na kilku rodzajach materiałów. Najczęściej używanym klasyfikatorem był wariant ?CNN?, oparty na wstępnie nauczonej sieci AlexNet.

Problemem przy porównywaniu prac naukowych w tym temacie jest często brak standardowej bazy zdjęć odpadów. Najbliższe temu jest ogólnie udostępniana baza \textit{ThrashNet} przez \textbf{YANG THUNG }. Była użyta w ?2-3? pracach. Poza tym zespoły najczęściej korzystają ze swoich własnych zdjęć. Wtedy występują różnice w rodzajach rozpoznawanych materiałów odpadów, ich stanie, samym sposobie fotografowania. Mając tak różne dane, ciężko porównać klasyfikator jednej grupy z drugą.

Dwa zespoły badaczy dokonały porównania metod SVM oraz CNN. Badacze z ?Lebabonu? stworzyli własną bazę ?XXXX zdjęć, z przedmiotów(plastikowych, metali oraz papierowych) [SAKR]. ?YT? połączyli zdjęcia odpadów z XXX z własnymi, tworząc XXX obiektów w ThrashNet w 6 klasach materiałów [YT]. Ich wyniki opisano poniżej.

SVM

SVM jest jednym z najbardziej podstawowych, prostych, ale też skutecznych klasyfikatorów. Metoda ta polega na podziale punktów danych na zbiory w wielowymiarowej przestrzeni. Nierozpoznany materiał jest przypisywany do zbioru z którym ma najwięcej wspólnych cech.

Zespół SAKR skorzystał z SVM z techniką BAG OF FEATURES. Ta technika polega na XXXXX. Nie dokonali redukcji zdjęć a klasyfikator wyćwiczyli w Matlabie, korzystając z 700 features, elo. Uzyskano dokładność 94.8 \%.

W zakresie rozpoznawania odpadów udało się uzyskać dokładność 94.8 \%, korzystając z SVM z techniką \emph{Bag of Features} \textbf{SAKR}.

Zespół YT wykorzystał SVM z XXXX oraz XXXX. Uzyskali 6000 % dokładności.

Także YANG THUNG uzyskali wynik dokładności 63 \%.

CNN

Wśród sieci neuronowych CNN jest najbardziej dedykowana do rozpoznawania zdjęć, w tym celu opracowana jest ich architektura. Wielu badaczy wskazuje na ich duży potencjał, jednak są też wymagające. Przede wszystkim należy przygotować jak największy ?dataset?. Zależnie od architektury, do ich nauki często wymagana jest moc obliczeniowa minimum kilku GB GPU. Ewentualnie można to skompensować redukując rozmiar zdjęć. Także, dobór HYPERPARAMETERS, czyli XXXXX, może zadecydować o sukcesie sieci.

AlexNet

ResearchNet

WaDaBa

Inne metody

Wspomaganie czujnikami, ZR

?Podsumowanie?

Autor wybrał bazę WaDaBa jako najbliższą tematowi pracy.

\section{Wstęp}

Ogólnie rozpoznawanie odpadów jest wymagające pod wieloma względami. Obiekty występują w każdym możliwym stanie: poziomie zgniecenia, zanieczyszczenia, kolorze, kształcie itd. Dodatkowo wewnątrz grup materiałów istnieje różnorodność. Chociażby rodzaj tworzych sztucznych PP (polypropylen) obejmuje zarówno sztywne, okrągłe, opakowania po jogurtach, śmietanie, jak i cienkie folie po batonach.

Z tego powodu środowisko naukowe oraz przemysłowe korzysta z metod uczenia maszynowego, a w szczególności uczenia głębokiego do segregacji odpadów.

Problemem przy porównywaniu prac naukowych jest często brak standardowej bazy odpadów. Najbliższe temu jest udostępniana baza \textit{ThrashNet} przez \textbf{YANG THUNG }. Była użyta w 2-3 pracach. Poza tym zespoły korzystają najczęściej ze swoich własnych zdjęć. Wtedy są różnice w rodzajach rozpoznawanych materiałów odpadów, ich stanie, samym sposobie fotografowania.

\section{SVM}

SVM jest jednym z najbardziej podstawowych, prostych, ale też skutecznych klasyfikatorów. Metoda ta polega na podziale punktów danych na zbiory w wielowymiarowej przestrzeni. Nierozpoznany materiał jest przypisywany do zbioru z którym ma najwięcej ,,wspólnego''.

W zakresie rozpoznawania odpadów udało się uzyskać dokładność 94.8 \%, korzystając z SVM z techniką \emph{Bag of Features} \textbf{SAKR}.

Także YANG THUNG uzyskali wynik dokładności 63 \%.

Oba zespoły badawcze jednocześnie próbowały z CNN i poszło im o wiele gorzej. Jest to jednak zw. przede wszystkim z brakiem wystarczająco dużej bazy danych lub mocy obliczeniowej. Wierzą że pokonując takie bariery CNN może przewyższyć SVM.

\section{CNN}

Duży potencjał, elastyczność. W praktyce wymagający pod względem rozmiaru \textbf{dataset}, GPU, odpowiednich \textbf{hyperparameteres}

AlexNet

\subsection{Z czujnikami}

Multilayer Hybrid Deep-Learning Method for Waste

Classification and Recycling

% \section{Inne}

% Histogram

\section{Podsumowanie}

Większość znalezionych materiałów wykorzystywało głębokie uczenie, głównie CNN. SVM jest pewnym algorytmem który stabilnie pozwoli uzyskać wyniki, jednak to CNN umożliwia uzyskanie, a nawet przewyższenie ludzkiej inteligencji.

% \section{Comparing Deep Learning And Support Vector

% Machines for Autonomous Waste Sorting -- Sakr}

% Dokonano porównania CNN oraz SVM. Badano 3 kategorie: plastik, papier, metal, używając kolorowych obrazów 256 x 256.

% SVM uzyskano dokładność 94.8 \%, zaś CNN 83 \%. Zauważono też, że SVM niezwykle dobrze dostosowało się do różnych typów odpadów. Do nauki CNN wykorzystano NVIDIA DIGITS zaś do SVM Matlab 2016a.

% \subsection{Porównanie metod}

% CNN, szczególnie w ostatnich latach, ma dobre predyspozycje do rozpoznawania obrazów 2D. Z drugiej strony SVM jest bardzo stabilnym klasyfikatorem. Zawsze udaje mu się rozpoznać dane, z największym marginesem błędu. W celu lepszych wyników, SVM można połączyć z CNN lub techniką \textbf{\emph{bag of features}}. W tej pracy postanowiono porównać CNN AlexNet i SVM z \textbf{\emph{bag of features}}.

% Po przeprowadzeniu badania autorzy twierdzą, że mając więcej dostępnego GPU oraz danych, CNN wyprzedzi SVM.

% \subsection{Użyte dane}

% Zebrano 2000 zdjęć, po 666 na materiał. 60 \% Służyło do nauki, 20 \% walidacji modelu, a 20 \% do ostatecznego testu modelu.

% W celu zapobiegnięciu przeuczeniu \textbf{\emph{dataset}} zwiększono do 6000 zdjęć zmieniając poziomy intensywności kanałów RGB.

% Z powodu ograniczeń GPU, przy treningu CNN obrazu zredukowano z rozmiaru 256 do 32. Wpłynęło to na dokładność modelu. Przy SVM to ograniczenie nie dotyczyło.

% Także, obliczono oraz odjęto obrazek który stanowi średnią wszystkich. Dzięki temu uzyskano normalizację jasności. Kiedy dwa obrazki są takie same, ale różnią się jasnością, to zostaną uznane za te same. Ma to poprawić dokładność detekcji.

% \subsection{Symulacja oraz wyniki}

% Badania przeprowadzono na komputerze z 2 GB GPU. Wykorzystano bibliotekę ,,Caffe''.

% \subsubsection{AlexNet}

% W celu optymalizacji wag użyto algorytmu \textbf{,,mini batch gradient descent backpropagation''}. GPU pozwoliło na użycie \textbf{\emph{batch}} o rozmiarze 100 obrazów. Zazwyczaj większy \textbf{\emph{batch}} pozwala na lepsza dokładność oraz zbieżność. Współczynnik uczenia wynosił $\alpha = 0.01$. Wyniki sieci przedstawiono na rys. \ref{rys:SakrConv} Udało się uzyskać dokładność 89 \% na zestawie walidacji. Także, różnica 0.2 w \textbf{\emph{loss}} pomiędzy danymi trenowania, a walidacji wskazuje na przeuczenie sieci. Można temu zapobiec podając więcej danych, lub zwiększając rozdzielczość obrazów(tu trzeba było ją zredukować). Ostatecznie na zestawie testowym uzyskano dokładność 83 \%.

% \begin{figure}[H]

% \centering

% \includegraphics[width=0.75\textwidth]{Sakr\_convergence.png}

% \caption{Wyniki dla CNN}

% \label{rys:SakrConv}

% \end{figure}

% \subsubsection{SVM}

% Użyto funkcji Matlaba ,,bagOfFeatures'', korzystająca z zestawu treningowego. Parametr \emph{K-means}, wyznaczający ilość grup, badano pomiędzy 400, a 600. Najdokładniejszy wynik na zestawie walidacji uzyskano dla wartości 500. Trening SVM przeprowadzono korzystając z funkcji ,,trainImageCategoryClassifier''. Dla zestawu testowego uzyskano dokładność 95 \%.

% \section{Classification of Trash for Recyclability Status -- Yang}

% \subsection{Porównanie metod}

% \subsection{Użyte dane}

% 6 różnych materiałów, łącznie 2400 zdjęć. Dokonano losowych rotacji, zmian jasności, translacji, skalowania, mapowania(\textbf{shearing}). Także dokonano odjęcia średniej oraz normalizacji (\textbf{mean subtraction and normalization})

% \subsection{AlexNet}

% Do CNN skorzystano z wstępnie nauczonej AlexNet i uzyskano średnio 88 \%. We use the Torch7 framework for Lua to construct our CNN. We implemented an eleven layer CNN that is very similar to AlexNet. Our network is smaller than AlexNet (using 3/4 of the amount of filters for some convolutional layers) because of computational constraints.

% The CNN was trained with a train/val/test split

% of 70/13/17, an image size of 256x256, 60 epochs,

% a batch size of 32, a learning rate of 5e-8, 5e-1

% weight decay every 5 epochs, an L2 regularization

% strength of 7.5e-2, and Kaiming weight initialization

% The network seemed

% to not learn, as the test accuracy we achieved in the

% experiment described was only 22 \%. the hyperparameters are not working well, or the model is too complex or too simple.

% \subsection{SVM}

% The SVM was chosen because it is considered one

% of the best initial classification algorithms and is not

% as complicated compared to a CNN.

% For a multiclass SVM, a common method is a one versus all classification where the class is chosen based on which class model classifies the test datum

% with greatest margin.

% Badacze wykorzystali SVM z \textbf{SIFT feauters}. On a high level, the SIFT algorithm finds blob like features in an image and describes each in 128 numbers. SIFT features are powerful because they are invariant to scale, noise and illumination which is perfect for recycling classification..

% Then bag of features was applied. The SIFT descriptors for the training images were clustered by the k-means algorithm where k was the number of training examples. Then for each new test example, the SIFT features are pulled and a histogram of values based on the original clustering is used as the data point for the dataset. This greatly reduces the required SVM training time since an image is reduced to a histogram.

% For the SVM, a radial basis kernel was chosen. Radial basis kernels are often the best for image datasets. The C parameter of the SVM was set to 1000. Gamma was set to an intermediate value of 0.5.

% It achieved a test accuracy of 63\% using a 70/30

% training/testing data split. The training error was

% 30 \%.