**­­­­­­­Sztuczna inteligencja w rozpoznawaniu obrazów. Od dopasowywania figur do sieci konwolucyjnych.**

**Wprowadzenie**

Wzrok jest tym zmysłem, który dostarcza nam więcej informacji niż wszystkie inne zmysły razem. Ze wszystkich wizualnych informacji dostępnych w otoczeniu około 100 Gbps pada na siatkówkę oka. Nie jest ona jednak wstanie aż tyle przyjąć ze względu na ograniczoną ilość i szybkość działania zakończeń nerwowych (ok. 137.000.000 wykrywających kontrast i ok. 5.000.000 wykrywających kolor), co uwzględniając ruchomość oka daje rozdzielczość siatkówki ok. 570 MP, przy czym jednocześnie efektywnie wykorzystywane jest zaledwie ok. 75 MP, a prędkość przetwarzania to do 90 klatek na sekundę, przy czym w punktach dużej koncentracji wzroku spada ona nawet do 4 klatek/s. Strumień informacji wychodzący z siatkówki to ok. 10 Mbps, a strumień wychodzący z całego oka to tylko ok. 10 Kbps i jest dalej redukowany w kolejnych etapach przetwarzania. Strumień informacji bezpośrednio wykorzystywany do świadomego podejmowania decyzji jest rzędu kilku do kilkudziesięciu bps (dokładne wartości zależą tu od tego, jak go zdefiniujemy).

Opis ten stanowi jednocześnie podpowiedź w sprawie rozwiązania architektury systemu rozpoznawania obrazów opartego na sztucznej inteligencji, której zamysłem jest wzorować się na inteligencji człowieka. System taki powinien również składać się z kolejnych warstw, a każda następna warstwa powinna przetwarzać mniejszą ilość informacji, jednocześnie na coraz wyższym poziomie abstrakcji. Tak też wygląda ogólny schemat systemu rozpoznawania obrazów przedstawiony na poniższym rysunku.



Rys. 1. Etapy systemu rozpoznawania obrazów.

Przy czym schemat taki może być realizowany albo przez poszczególne wyspecjalizowane bloki, albo np. przez odnoszącą w ostatnim czasie wielkie sukcesy konwolucyjną sieć neuronową, która już sama realizuje większość tych etapów i w związku z tym nie jest od nas wymagane ich szczegółowe zaprojektowanie.

Typowe zadania komputerowego przetwarzania obrazów to rozpoznawanie obiektów na obrazie (zwane skrótowo rozpoznawaniem obrazów), analiza ruchu (np. śledzenie obiektów), rekonstrukcja scen 3D ze zdjęć, poprawa jakości obrazów (np. redukcja szumów). Na wykładzie skoncentrujemy się na rozpoznawaniu obiektów na obrazie. Klasycznym problemem jest tu stwierdzenie, czy dane zdjęcie zawiera obiekt określonego typu, o określonych cechach lub wykonujący określoną aktywność (np. czy na zdjęciu jest człowiek/pies/samochód i co robi).

Omówione zostaną cztery grupy algorytmów rozpoznawania obrazów: 1) metody dopasowania figur i inwariantne, 2) metody ruchomego okna, 3) deskryptory cech, 4) konwolucyjne sieci neuronowe. Można w uproszczeniu przyjąć, że grupy te kolejno charakteryzują się coraz większą złożonością obliczeniową, natomiast pozwalają na osiągnięcie coraz bardzie spektakularnych efektów. Dlatego też powinny zostać odpowiednio dobrane do danego zagadnienia. W prezentacji każdej kolejnej grupie będzie poświęcone coraz więcej czasu. Najdokładniej omówione zostaną metody wykorzystujące głębokie architektury sieci neuronowych, które aktualnie umożliwiają uzyskanie największych dokładności w rozpoznawaniu obrazów.

**Metody dopasowania figur i inwariantne**

**Pierwsza generacja – dopasowanie figur.** Początki prac nad automatycznym rozpoznawaniem liter sięgają już końca XIX stulecia. Około roku 1920 Emanuel Goldberg stworzył pierwszą maszynę, która odczytywała znaki i przetwarzała je na kody telegrafu. W 1954 w Rider Digest był już używany prawdziwy system OCR (optical character recognition). W systemach tych stosowano dwa rodzaje algorytmów rozpoznawania obrazów: porównywanie litery do wzorca piksel po pikselu oraz ekstrakcja cech jak linie, zamknięte pętle, itp. i porównywanie ich do cech wzorca. Inne przykłady to inspekcje kształtu przedmiotów, który powinien być prostokątem, lub kołem
i metody rozpoznawania, czy dany przedmiot rzeczywiście taki jest w oparciu
o dopasowanie figur poprzez zliczanie jaka część pikseli znajduje się poza żądanym zakresem. Kolejnym etapem było zastosowanie operacji morfologicznych jak erozja i dylatacja do wyodrębnienia ze zdjęć interesujących obiektów i ich detekcji na podstawie odmiennego koloru. Np. E. R. Davis zastosował to w praktyce do wykrywania zanieczyszczeń w ziarnach zboża.

**Druga generacja – metody inwariantne.** Wadą metod pierwszej generacji było to, że jeżeli obiekt był rozpoznawany na podstawie kształtu, a nie koloru, to gdy zmienił swój rozmiar lub został obrócony o pewien kąt to często nie był już poprawnie identyfikowany. Na początku lat 1990-tych zaproponowano kilka metod inwariantnych, np. system LEWIS Rothela, polegający na wyodrębnieniu z obiektu charakterystycznych fragmentów (np. linie, łuki) i opracowaniu statystyk ich wzajemnego położenia w zbiorze obrazów. W późniejszych pracach cechy były podobnie wyodrębniane, lecz do rozpoznania obiektu zamiast statystyk została wykorzysta klasyfikacja
z użyciem różnych algorytmów sztucznej inteligencji. O ile część klasyfikacyjna działała dość dobrze to problemem był dobór odpowiednich cech. Znaczną poprawę odniesiono tu stosując algorytmy następnej generacji oparte na deskryptorach cech.

**Metody ruchomego okna - Detekcja twarzy**

W 2004 roku Paul Viola i Michael Jones opublikowali artykuł “Robust Real-Time Face Detection”. Zaprezentowali w nim algorytm detekcji twarzy, który jak się okazało działał tak dobrze, że do tej pory jest w zasadzie standardem de facto w detekcji twarzy na zdjęciach. Metoda ta zawdzięcza swój sukces względnej prostocie, szybkości i jednocześnie dokładności działania.

 

Rys. 2. Po lewej: cechy w algorytmie Viola-Jones: po prawej: wyrównywanie cech z twarzą.

Aby uprościć zadanie autorzy ograniczyli się do rozpoznawania twarzy ustawionej na wprost
w kierunku kamery. Jednym z podstawowych problemów było tutaj to, że zarówno wielkość twarzy, jak i jej pozycja na zdjęciu może być różna. Algorytm Viola-Jones przemieszcza ruchome okno po całym zdjęciu, poszukując w nim twarzy, przy czym okno to jest skalowane do różnych wielkości i różne wielkości tego okna są kolejno przesuwane przez całe zdjęcie. Jako, że jest to detektor niezależny od skali, to ilość wymaganych obliczeń jest przy tym niezależna od wielkości okna. Detektor Viola-Jones analizuje poszczególne okna sprawdzając występowanie w nich cech złożonych z dwóch lub więcej prostokątów, ciemnych i jasnych. Większość obszaru zdjęcia jednak nie będzie zawierać twarzy. Korzystając z tego można działanie algorytmu podczas predykcji przyspieszyć, dokonując najpierw wstępnej, szybszej predykcji - jeśli stwierdzi ona, że badane okno na pewno nie zawiera twarzy, od razu przechodzi się do kolejnego okna. Jeśli stwierdzi, że może zawierać twarz, to dokonywana jest predykcja przez kolejny klasyfikator .



Poprzednie systemy osiągnęły dokładność rozpoznawania twarzy między 85% a 95% i niezwykle mało false positives – rzędu 10exp(-5)-6. Zadaniem kaskady klasyfikatorów jest więc utrzymanie takich parametrów dla każdego stopnia. Końcowa dokładność detekcji takiej kaskady jest więc znacznie większa, niż dokładność jej poszczególnych stopni. Końcowa ilość fałszywych rozpoznań jest również większa, ale przy takich współczynnikach true positeves i false positives opłaca się zrobić kaskadę – nie tylko ze wgl. na dokładność ale i na minimalizację czasu detekcji.

Począwszy od 2007 roku zaczęto detekcję twarzy stosować w aparatach fotograficznych celem automatycznego ustawiania ostrości na ten punkt zdjęcia, w którym wykryto twarz.

**Autonomiczne samochody**

Rozpoznawanie obrazów w kierowaniu samochodem może służyć dwóm celom: 1. Pomoc kierowcy w bezpiecznym prowadzeniu samochodu. 2. Automatyczne kierowanie samochodu przez komputer.

Pomoc dla kierowcy jest zagadnieniem wymagającym mniejszej mocy obliczeniowej i aparatury,
a więc możliwym do łatwego zaim­­plementowania i już stosowanym niektórych seryjnych samochodach. Zadaniem takiego systemu będzie: obserwacja oczu kierowcy, aby stwierdzić, czy nie zasypia, detekcja przekroczenia linii rozdzielającej pasy jezdni bez włączonego kierunkowskazu, wykrywanie świateł i znaków drogowych, pieszych na jezdni, zbyt małej odległości między samochodami i informowanie kierowcy o tych zdarzeniach.

  

Rys. 3. Po lewej: Google Self-Driving Car, po prawej: Junior z Uniwersytetu Stanford.

W pełni autonomiczny samochód poruszający się bez kierowcy jest sprawą bardziej skomplikowaną. Tego typu projekty wykorzystują na ogół podobne sposoby rozwiązań. Omówimy je na przykładzie systemu sterowania samochodu rozwijanego najpierw jako projekt Junior na uniwersytecie Stanford przez zespół kierunkiem Sebastiana Thruna a aktualnie kontynuowanego jako Google Self Driving Car. Samochód ma zamontowany komplet czujników, a w bagażniku komputer sterujący jego pracą. Cena tego wyposażenia nawet 10-krotnie przekracza wartość samego samochodu. Aktualnie Google Car dodatkowo wykorzystuje bezprzewodowe połączenie
z klastrem obliczeniowym wspomagającym online jego sterowanie. Celem wprowadzenia automatycznego sterowania samochodu do seryjnej produkcje należy więc nie tylko zmniejszyć jego niezawodność, ale i zdecydowanie ograniczyć koszty. 40 km/h

Przy kierowaniu samochodem występują tu dwa zagadnienia:

1. Określenie dokładnej lokalizacji samochodu względem otoczenia i tu wykorzystywany jest system radarowy LIDAR wspomagany przez GPS i opcjonalnie zdjęcia z Google Street View.

2. Identyfikacja obiektów dynamicznych jak piesi, inne pojazdy, kolor świateł na skrzyżowaniach.

System LIDAR dostarcza trójwymiarowe mapy w podczerwieni, więc umożliwia tworzenie map także w nocy. Rozdzielczość otrzymanej mapy wynosi 5 cm, zaś dokładność sterowania samochodem 10 cm. GPS jest używany jako ograniczenie, celem zapobieżenia zbyt dużych błędów w lokalizacji na wypadek niewłaściwego zadziałania algorytmu określania położenia samochodu. Występuje konieczność jednoczesnej generacji mapy i określania położenia samochodu wzgl. tej mapy. Używane są tu różne algorytmy do określenia współrzędnych x, y oraz kierunku jazdy samochodu, np. Monte Carlo Localizer.

**Deskryptory cech**

Deskryptory cech można traktować jako systemy rozpoznawania obrazów czwartej generacji. Ich działanie wzorowanie jest na sposobie działania człowieka przy układaniu puzzli. Patrząc na niektóre fragmenty układanki od razu jesteśmy w stanie określić gdzie powinny być. To przeważnie rogi czegoś, lub jakiś inny punkt zdecydowanie wyróżniający się z otoczenia. Położenie innych elementów jesteśmy w stanie określić w przybliżeniu – to krawędzie obiektów, natomiast położenia elementu we wnętrzu jednobarwnego obiektu nie będziemy w stanie określić. Na tej samej zasadzie działają deskryptory cech: starają się znaleźć najbardziej znaczące cechy dla danego rodzaju obiektu – czyli jego rogi, lub ogólnie takie punkty, które nawet przy niewielkim przesunięciu
w którąkolwiek stronę przestają pasować do obrazu.

 

Rys. 4. Po lewej: cechy charakterystyczne, po prawej: detekcja FAST. (źródło: [docs.opencv.org](http://www.opencv.org))

Na tej zasadzie powstał detektor rogów Harrisa. Oblicza on o ile się zmieni zawartość danego okna przy przemieszczeniu go we wszystkich kierunkach – czym więcej, tym jest to bardziej istotny punkt, przy czym przyjmuje się pewną progową wartość zmiany, powyżej której uznaje się go za punkt charakterystyczny. Detektory te są inwariantne dla rotacji, to znaczy dokonując obrót obrazu dalej wykrywają te same punkty. Nie są jednak inwariantne dla skali, gdyż nie można używać tych samych rozmiarów okna dla wykrycia różnej wielkości rogów.

Zaproponowany w 1999 roku SIFT – Scale Invariant Feature Detector rozwiązuje ten problem poprzez wykrywanie punktów charakterystycznych o różnych rozmiarach. Następnie SIFT przypisuje orientację poszczególnym punktom, aby zapewnić inwariantność względem rotacji. Za tymi wszystkimi operacjami stoją dość złożone, a zatem i czasochłonne obliczenia (rozmywanie obrazów z użyciem kerneli gausowskich, tworzenie piramid, histogramów w różnych kierunkach), dlatego z czasem opracowano bardziej wydajne obliczeniowo metody, zwłaszcza FAST w 2006 roku. FAST to skrót od Features from Accelerated Segment Test, ale może być to również rozumiane jako szybki. FAST jest również bardziej intuicyjny.

FAST używa okręgu złożonego z 16 pikseli na obwodzie i jeżeli *n* kolejnych pikseli na tym okręgu jest jaśniejszych od jego środka o co najmniej *p* lub ciemniejszych o co najmniej *p* (w skali szarości) to wówczas środek traktuje się jako róg (Rys. 4 po prawej).

Po wyznaczeniu kluczowych punktów na dwóch obrazach trzeba je dopasować, aby stwierdzić, czy obrazy te przedstawiają ten sam rodzaj obiektu, co znając jeden obiekt sprowadza się do rozpoznania drugiego obiektu. W rozpoznawaniu obrazów ten ostatni etap sprowadza się do zagadnienia klasyfikacji, gdzie wydobyte z obrazów cechy poddaje się odpowiednim przekształceniom (np. z wykorzystaniem klasteryzacji i histogramów), aby utworzyć z nich zbiór atrybutów, na których uczony jest model predykcyjny. Tym modelem był przeważnie SVM, gdyż jeszcze kilka lat temu dawał on często największą dokładność klasyfikacji dla tego typu zagadnień, ale można oczywiście stosować różne modele, w tym komitety modeli.

W kolejnych latach opracowano kilka udoskonaleń, z których najbardziej istotny wpływ na dziedzinę miały Visual Bag of Words (2003) oraz HOG (2008). Ponieważ SIFT nie zawsze był wstanie znaleźć dopasowanie między tym samym rodzajem obiektu, ale różnie zdeformawanym na dwóch zdjęciach (np. to samo zwierzę siedzące i biegnące), więc powstała konieczność zakodowania faktu, że poszczególne części obiektu mogą zajmować różne położenie względem siebie, a niektóre fragmenty mogą być niewidoczne. Zatem wprowadzono słownik wizualny (Visual Dictionary) otrzymywany przez wykonanie klasteryzacji na cechach SIFT (lub innego deskryptora), która odwzorowuje 128-wymiarowy rzeczywisty wektor na liczby naturalne będące centrami klastrów. Histogram takich wizualnych słów stanowi reprezentację dobrze radzącą sobie z różnymi deformacjami rozpoznawanych obiektów.

W 2008 roku wprowadzono detektor cech zwany HOG 2008 (histogram of oriented gradients), który od SIFT różnił się tym, że dokonywał operacji w przesuwnym oknie (podobnie, jak algorytm Viola-Jones). HOG szuka wystąpienia cech w poszczególnych oknach (częściach obrazu), co powoduje jego zwiększoną dokładnością.

**Głębokie uczenie - konwolucyjne sieci neuronowe**

To już metody piątej generacji

Dotychczasowe metody: ekstraktora cech i klasyfikator.

Ekstrakcja cech zawierała w sobie wiedzę o zjawisku .

wzrost mocy + duże zbiory

W 1998 roku Yann Lecun wraz ze współautorami opublikowali artykuł „Gradient-Based Learning Applied to Document Recognition” – jak deskryptory cech, dobre wyniki 2012

2-wymiarowe sąsiedztwo, niedostosowanie NN, brak inwariancji lub olbrzymia sieć

W odpowiedzi na nowe możliwości (duża moc obliczeniowa, dostępne duże zbior) oraz na przedstawione braki tradycyjnych sieci neuronych zaproponowano sieci konwolucyjne, który łączą trzy idee:

1. lokalne sąsiedztwo punktów obrazu,
2. współdzielone wagi, czyli zmniejszenie liczby parametrów oraz
3. sub-sumblig czyli stopniowe zmniejszanie rozdzielczości.

To już metody piątej generacji – aktualnie najdokładniejsze w rozpoznawaniu obiektów na zdjęciach. Metody, które omawialiśmy w poprzednich częściach wykładu są nadal stosowane i nie wydaje się celowe, żeby wszędzie próbować zastosować sieci konwolucyjne, zwłaszcza
w zagadnieniach, gdzie istniejące rozwiązania się dobrze sprawdzają. Choć również inne architektury sieci głębokich dają dobre wyniki w rozpoznawaniu obrazów, to umówimy tu tylko sieci konwolucyjne jako aktualnie dominujący trend. W porównaniu do klasyfikacji obrazów
z wykorzystaniem deskryptorów cech są one nie tylko bardziej dokładne, ale i prostsze
w implementacji. To, że sieci te nie wymagają od nas wiedzy o problemie i wstępnego przygotowania danych jest również ich znaczącą zaletą.

Dotychczas stosowane metody rozpoznawania obrazów składały się z dwóch głównych modułów: ekstraktora cech i klasyfikatora. Ekstraktor cech transformował właściwości poszczególnych pikseli tak, aby było reprezentowane przez nisko wymiarowe wektory, tak, aby z jednej strony było one inwariantne względem transformacji i zniekształceń, a z drugiej strony były łatwe do porównywania przez klasyfikatory.

Ekstrakcja cech zawierała w sobie wiedzę o zjawisku, np. to, że do ułożenia puzzli potrzebujemy znaś głównie lokalizację rogów, to, jak rogi sa zdefiniowane, itd. Zaprojektowanie takiego systemu wymagało dużo pracy. Aby to zobrazować może pokażę tu jeszcze raz, jak gruba jest książka profesora Daviesa traktująca właśnie głównie o projektowaniu cech. Oczywiście książka ta nie wyczerpuje jeszcze całego tematu. Natomiast klasyfikator używany w drugim module był klasyfikatorem ogólnego zastosowania, jak np. sieć neuronowa, SVM albo random forest.

Dużym problemem takiego podejścia było to, że dokładność rozpoznawania była w znacznym stopniu zdeterminowana poprzez umiejętność zaprojektowania odpowiedniego zestawu cech.

Wraz ze wzrostem mocy obliczeniowej komputerów konieczność operowania na niskowymiarowych przestrzeniach cech zaczęła tracić na znaczeniu i stało się możliwe operowanie bezpośrednio na wysokowymiarowych przestrzeniach pikseli. Drugą zmianą, jaka się dokonała, było pojawianie się zoraz większej ilości coraz większych zbiorów różnych obrazów. Jest to istotne, ponieważ, jeżeli model ma operować na wysokowymiarowej przestrzenii danych, to musi mieć dużo parametrów ustalanych w procesie uczenia. A z kolei, aby dużą ilość paramtrów można było nauczyć, musi byż dużo wektorów uczących. Zaistnienie tych dwóch czynników stworzyło grunt do podjęcia prób utworzenia w pełni autonomicznego systemu rozpoznawania obrazów, który zapewni zarówno automatyzację projektowania cech, jak i klasyfikację. Podjęto oczywiście różne próby i zbudowano kilka modeli. Modelem, który odniósł największy sukces były konwolucyjne sieci neuronowe. Przy czym jako sukces rozumiemy tu zarówno dokładność rozpoznawania obrazów, jak i umiejętność wypromowania tego modelu. Nie można wykluczyć, że ktoś opracował lepsze rozwiązanie, jednakże nie potrafił, albo nie chciał go wypromować.

W 1998 roku Yann Lecun wraz ze współautorami opublikowali artykuł „Gradient-Based Learning Applied to Document Recognition”, w którym zaprezentowali konwolucyjną sieć neuronową nazwaną LeNet5. Co ciekawe sieć ta początkowo jeszcze nie odniosła tak spektakularnych sukcesów i jakość jej działania była porównywalna z jakością metod opartych na deskrytporach cech, ale stała się ona początkiem prac nad metodami nowej generacji. W wyniku kolejnych badań różnych autorów wprowadzano do tej sieci kolejne ulepszenia i aktualne wersje działają zedcydowanie lepiej od deskryptorów cech. Zacznijmy jednak od omówienia oryginalnej sieci LeNet5.

Sieć LeNet5 została początkowo wykorzystana do rozzpoznawania pisma ręcznego: odręcznie pisanych cyfr. Autorzy najpierw zauważyli niedostosowanie istniejących klasyfikatorów do tego zadania, ponieważ traktowały one poszczególne cechy jako niezależne. Natomiast w przypadku obrazów istnieje oczywiste sąsiedztwo pikseli i daną cechą są nie tyle właściwości pojedynczego piksela, co zależności między pojedynczym pikselem a jego sąsiadami. Należy więc odwzorować to 2-wymiarowe sąsiedztwo w obrazie na 2-wymiarowe sąsiedzwtwo w modelu rozpoznającym ten obraz. Kolejnym problemem jest fakt, że przy w pełni połączonych warstwach bardzo szybko rośnie ilość wag sieci, których wartości trzeba ustalać w procesie uczenia, a więc rośnie i złożoność obliczeniowa. Jeszcze jednym problemem tradycyjnych sieci neuronowych jest to, że nie posiadają wbudowanego mechanizmu inwariancji dla różnych przeksztalceń geometrycznych jak rotacja, skalowanie i przesunięcie. Choć taka inwariancje byłaby możliwa do uzyskania, to wymagałaby ogromnej liczby parametrów w sieci.

 W odpowiedzi na nowe możliwości (duża moc obliczeniowa, dostępne duże zbior) oraz na przedstawione braki tradycyjnych sieci neuronych zaproponowano sieci konwolucyjne, który łączą trzy idee:

1. lokalne sąsiedztwo punktów obrazu,
2. współdzielone wagi, czyli zmniejszenie liczby parametrów oraz
3. sub-sumblig czyli stopniowe zmniejszaie rozdzielczości.

Każdy neuron w pierwszej warstwie ma podłączone wejścia do zbioru 25 pikseli znajdujących się w pobliżu. W ten sposób może on dokonać ekstrakcji elementarnych cech. Te cechy są nastepnie formowane w cechy wyższego rzędu w kolejnych warstwach sieci. Najniższy rząd cech, to np. krawędzie, wyższy to wierzchołki, kończe odcinków, jeszcze wyższe to poszczególne cześci obiektów, np. łapy psa, czy koła samochodu, aż wreszcize na najwyższym poziomie rozpoznawane są całe obiekty. Detektory elementarnych cech przydatne w jendym fragmencie obazu będą przydatne rowież w jego zopostałych częściach. Ta cecha została wykorzystana do zmuszenia neuronów,których pola recepcyjne są zlokalizowanie w różnych częściach obazu do współdzielenia wag, czyli do tego, aby miały identyczne zestawy wag. Neurony w ramach warstwy są zorganizowane w podgrupy, w ramach których wszystkie neurony mają jednakowe zestawy wag. Zbiór wyjść z wektorów z takiej podgrupy nazywany jest mapą cech (feature map). Wektory w mapie cech wykonują te same operacje na różnych cżęściach obrazu. Cała warstwa konwolucyjna składa się z kilku map cech, co umożliwia ekstrakcję kilku różnych cech w każdej części obrazu. Współdzielenie wag dodtakowo zapernia zmniejszenie złożoności obliczeniowej modelu. W przypadku sieci LeNet5 z 341 tys. do 60 tys. wag, a więc ok. 6-krotne. Dodatkowo zapewnia to także lepszą generalizację sieci, poprzez ograniczenie jej przeuczania.

Cyfry w zbiorze MNIST były zeskanowane i zapisane w rozdzielczości 32x32 piksele. Przy czym sama cyfra zajmowała maksymalnie 28x28 pikseli, zaś wolnie miejsce po bokach słyżyło do tego, że można było dokonać detekcji cech, żeby była jeszcze zapewniona różnica kolowu między cyfrą, a tłem z każdej strony.

każdy neuron w mapie cech ma 25 wejść połączonych do obszaru 5x5 pikseli. Sygnałami przekazywanymi do tych wejść jest stopień szarości poszczególnych pikseli. Zaś w przypadku obrazów kolorowych mielibyśmy w pierwszej warstwie sieci trzy zestawy map cech, każda dla jednej składowej RGB i syg onałami były by wartości tych składowych. W drugiej warstwie te sygnały byłyby już scalane razem. Choć są oczywiście możliwe także inne rozwiązania architetoniczne sieci nokwolucyjnych do rozpoznawania kolorowych obrazów.

Pola recepcyjne sąsiednich neuronów się częściowo nakladają na siebie. Jest to zrobione celem zapewnienia ciągłości analyzy obrazu, żeby nie zostaly pominięte cechy leżące na granicy dwóch różnych pól recepcyjnych, gdyby te się nie nachodzily na siebie.

Ponieważ mamy tu 6 map cech, więc w każdej części obrazu możemy wykrywać 6 różnych cech. Jak cechy już zostaną wykryte, to ich dokładna lokalizacja staje się mniej istotna, jedynie położenie jednych cech względem innych jest istotne. Chodzi bowiem o wykrycie co jest na zdjęciu, czyl
w tym przypadku jaka cyfra, a nie w ktorym miejscu zdjecia się obiekt znajduje. Dokładna pozycja poszczególnych cech nie tylko jest nieistotna, ale wręcz niepożądana, gdyż powodowałoby to uczenie się przez sieć rzeczy, które nie mają wpływu na to jaka jest to cyfra i w efekcie utrudniało nauczenie się tego, co powinna. Prostym sposobem na ograniczenie dokładności lokalizacji cech jest zmniejszenie wymiarowości i temu służy następna warstawa typu sub-sampling, która zbiera sygnały z kwadratu o wymiacha 2x2 nurony i je uśrednia, a następnie mnoży przez pewien współczynnik ustalany w procesie uczenia sieci i dodaje bias również ustalany w procesie uczenia. Całość jest przyhształcana przez sigmoidalną funkcjetransferu. Tu już nie ma nachodzenia na siebie poszczególnych kwadratów, tak, jak było w piewszej warstwie. W efekcie dostajemy mapy cech
o zerdukowanej wymiarowości o rozmiarhach 14x14.

Kolejne dwie warstwy C3 i S4 to znowu warstwa konwolucyjna i podpróbkująca. Działają analogicznie, jak dwie pierwsze warstwy. Warstwa C3 zawiera 16 map cech, każdy neuron w mapie cech ma 25 wejść połączonych do obszaru 5x5 neuronów z warstwy S2. Przy czym mapy cech
z warswty S3 nie są połączone do wszystkich map cech warstwy S2. Po pierwsze ogranicza to ilość parametrów całej sieci, a po drugie wymusza asymetrię i w ten sposób róźne mapy cech w warstwie C3 wydobywają różne cechy, które często się wzajemnie uzupełniają.

Ostatnie 3 warstwy mające architekturę standardowej w pełni połączonej sieci dokonują ostatecznej klasyfikacji. Może to być sieć zarówno typu MLP, jak i RBF jak i jakaś inna sieć zbudowana na bazie tych architektur. W przypadku sieci LeNet 5 warstwa F6 zawierała 84 neurony, w których funkcją traksferu był tangens hiperboliczny, a końcowa warstwa 10 neuronów RBF. W tym oryginalnym rozwiązaniu nie było tak, że każdy neuron RBF odpowiadał jednej cyfrze od 0 do 9, tylko klasa była określana na podstawie pewnej konfiguracji wyjść tych neuronów uwzględniającej pewno podobieństwo cech, ale to jest już szczegół implementacyjny i wcale tak być nie musi
i może być też każdy neuron wyjściowy reprecentować inną klasę, jak w tradycyjnych sieciach. Można powiedzieć, że w pewnym sensie te końcowe w pełni piłączone warstwy pełnią rolę klasyfikatora, natomiast poprzednie warstwy konwolucyjne i podprobkujące są ekstraktorem cech. W tym sensie sama idea się wiele nie zmieniła względem poprzednich rozwiązań, natomiast zdecydowanie zmieniła się implementacja, połączając sekwencyjne proocesy ekstrakcji cech
i klasyfikacji w jeden dokujący się jednocześnie zintegrowany proces i pozwalając sieci samodzielnie decydować, jakie cechy chce wykrywać i jak je następnie łączyć i przetwarzać.

**Googlenet**

Najbardziej oczywistym sposobem poprawy jakości CNN jest zwiększenie ich rozmiaru. Zarówno ilości warstw, jak i ilosci neuronów w warstwach. Jednakże wiąże się to z dwoma problemami: sieci z większa liczbą parametrów wymagają większych zbiorów uczących oraz większych nakładów obliczeniowych.

Na wejścia sieci podajemy bowiem bezpośrednio składowe RGB każdego piksela i uczymy sieć metodami znanymi a uczenia sieci neuronowych (gradient stochastyczny, Rprop lub inne metody), a ona już sama dokonuje poszczególnych operacji (zapewniając niezależność od przesunięcia, skali i obrotu), jak ekstrakcja cech na niższych poziomach i łączenie elementarnych cech w większe fragmenty na wyższych poziomach, aż ostatecznie dokonuje klasyfikacji. Co więcej sieć ta najczęściej jest w stanie wyodrębnić bardziej efektywny zestaw cech, na podstawie których uzyskuje lepsze wyniki klasyfikacji, niż te uzyskane przy pomocy deskryptorów cech. Być może
w przyszłości zostaną opracowane rozwiązania sieci również samodzielnie dobierających optymalną architekturę do danego zagadnienia, ale na razie samą strukturę sieci musimy jeszcze samodzielnie zaprojektować.

Wadą sieci konwolucyjnych jest natomiast duży nakład obliczeniowy potrzebny na nauczenie sieci, w niektórych przypadkach nawet tygodnie obliczeń na klastrach komputerowych, z użyciem wielu CPU i GPU. Dlatego też do niektórych zastosowań można ściągnąć z Internetu już nauczone sieci, gotowe do użycia, np. GoogLeNet, którą omówimy w dalszej części.

Rozwiązaniem problemu niewystarczającej ilości przypadków uczących i uzego nakladu obliczeniowego jest odejście z wpełni polączonej architektury na rzecz rzadko połączonych architektur. Ma to też takie uzasadnienie, że na zdjęciach w storunkowo niewielu miejscach są charakterystyczne punkty. Na podstawie przeprowadzonych badań wiadomo, że dystrybucja prawdopodobieństwa w wielkich zbiorach danych może być dobrze reprezentowana przez wielkie, lecz z rzadkimi połączeniami sieci neuronowe. Architekturą taka można stworzyć warstwę po warstwie poprzez połączenie w klastry neuronów z mocno skorelowanymi wyjściami.

Nadal pozostaje tu jednak pewnym problemem brak efektywnych algorytmów wykonywujących obliczenia ma macierzach rzadkich, choć zostały już podjęte prace koncentrujace się na klastrowaniu rzadkich macieszy w storunkowo gęste podmacierze, to efektywność tych metod jeszcze zostawia jeszcze dużo możliwości do poprawy. Na tej idei też w znacznej mierze oparte jest działanie sieci GoogLeNet. Ten zestaw warst wykonujących konowolucje o różnej wiejkości sluży wyłapaniu carakterystycznych punktów w różnej skali. Natomiast jego wersja pokazana po prawej stronie z dodatkowymi żółtymi warstwami służy redukcji wymiarowości, a więc i zmniejszeniu nakladu obliczeniowego. Cała sieć składa się z 9 takich warst 3-warstswywych i jeszcze dodatkowych warstw, co w sumie daje 22 warstwy.

Do uczenia sieci wykorzystano asynchroniczny gradient stochastyczny z momentum równym 0.9
i współczynnikiem uczenia zmniejszanym o 4% co 8 epok.

Ze względu na to, że sieć ma aż 22 warstwy uczone metodą gradientową, występuje już wyraźne zjawisko zanikającego gradientu. Jednakże ponieważ sieci o płytszych architekturach również dają całkiem dobre rezultaty, można zauważyć, że neurony w środkowych warswach sieci GoogLeNet już całkiem dobrze reprezentują informację potrzebną do klasyfikacji. Zatem dwa razy
w środkowych warstwach zostały dodane wyjścia, na których jest również liczony błąd i jest on propagowany wstecznie razem z błędem pochodzącym od głównego wyjścia. Każdy z tych błędów jest uwaględniany z pewną wagą, w przybliżeniu jednakową. Natomiast podczas precykcji już nauczonej sieci bierze sie pod uwage tylko jej wyjścia w ostatniej warstwie.

Zauważono, że z pewnymi typami problemów lepiej sobie radzi człowiek, a z innymi sieci konwolucyjne. Człowiek lepiej rozpoznaje małe i cienkie obiekty, np. łodygi kwiatów, mrówki, czy zniekształcone obiekty. Przypuszczalnie wynika to z doświadczenia człowieka w takich przypadkach – nie musi dokładnie widzieć takiego obiektu, by zgadnąć na podstawie jego zgrubnych zarysów co się może znajdować w danym miejscu i danej sytuacji. Natomiast sieci konwolucyjne radzą sobie lepiej z klasyfikacją podobnych zwierząt, jak np. różne gatunki ptaków lepiej niż ogół ludzkości, ale już nie lepiej niż specjaliści od ptaków. Tu znowu ludzie mają problem z wykryciem charakterystycznych cech danego gatunku, jeśli ktoś im tego nie powie, np. że decydujące są czarne końcówki skrzydeł, czy czarny kołnierz ptaka. Sieć konwolucyjna te cechy wykrywa i stwierdza różnicę.