2. Algorytmy detekcji twarzy

Na przestrzeni lat zaproponowano wiele różnych podejść do zagadnienia detekcji twarzy. Zagadnienie wykrywania twarzy wraz z opisem metod wykorzystywanych w przeszłości, tych obecnie najpopularniejszych oraz o metodach które mogą zostać wykorzystane w przyszłości przedstawiono w pracy [Stefanos Zafeiriou 2015]. Dodatkowo w publikacji [AL-Allaf 2014] zamieszczono szczegółowy przegląd systemów detekcji twarzy opartych o sieci neuronowe.

Można wyróżnić podział na algorytmy bazujące na:

- detekcji cech szczególnych twarzy takich jak symetria oczu, położenie nosa i ust oraz relacje pomiędzy nimi,
- teksturze, działające niezależnie od oświetlenia i pozycji obiektu,
- porównywaniu wzorców i szukaniu elementów wspólnych pomiędzy obrazem zapisanym w pamięci, a badaną sceną,
- metodach opartych na uczeniu maszynowym,
- informacji o kolorze,
- relacji pomiędzy kolejnymi scenami – twarz wykrywana w sekwencji obrazów (przetwarzanie wideo), wykorzystanie informacji o kolorze oraz krawędziach.

Możliwe jest łączenie kilku metod, co może skutkować znacznym podniesieniem wskaźnika skuteczności detekcji. Większość obecnie stosowanych rozwiązań skupia się na przypadku, w którym twarz skierowana jest wprost do kamery, a warunki oświetleniowe są niemal idealne (obiekt dobrze oświetlony, zauważalny kontrast pomiędzy poszczególnymi częściami sceny).

Ważnym aspektem dotyczących realizacji algorytmów detekcji twarzy jest sprawdzenie ich skuteczności oraz ocena złożoności obliczeniowej – w szczególności, czy możliwa jest implementacja systemu działającego w czasie rzeczywistym. Za system czasu rzeczywistego można przyjąć system, w którym nie występuje utrata informacji spowodowana zbyt wolną jego pracą [Gorgoń 2013]. Przykładem może być detekcja dla każdej kolejnej przychodzącej klatki materiału video.

Można zaobserwować zależność pomiędzy stopniem skomplikowania algorytmu, a skutecznością detekcji, gdzie jego większa złożoność najczęściej przekłada się na bardzo wysoką skuteczność. Aby lepiej zrozumieć problem detekcji i rozpoznawania twarzy warto opisać podstawowe problemy, z którymi trzeba się zmierzyć.

- Pozycja – twarz na zdjęciu nie zawsze znajduje się w pozycji na wprost do kamery, a cechy szczególne twarzy mogą być częściowo lub całkowicie niewidoczne np. częściowe przesłonięcie twarzy przez padający cień.

- Oświetlenie i warunki, w których pracujemy – źródło światła wpływające na postrzeganie koloru przez kamerę – balans bieli, intensywność światła.
- Zasłonięcie przez inne obiekty, np. okulary, broda, włosy,
- Różnice w budowie twarzy wynikające z pochodzenia oraz cechy indywidualne.

Czasami konieczne może być określenie dodatkowych warunków, w których pracować będzie urządzenie z zaimplementowanym algorytmem. Można do nich zaliczyć:

- maksymalną liczbę osób, która może zostać wykryta,
- częstotliwość z jaką detekcja ma być przeprowadzana – w sposób ciągły czy w określonych odstępach czasu,
- wielkość wykrywanych twarzy w kadrze.

Pomimo znacznych środków, jak też ogromnego zainteresowania środowiska naukowego i komercyjnego, temat ten z pewnością pozostanie jeszcze przez długi czas obszarem, w którym pojawiają się innowacyjne rozwiązania i przeprowadzane będą liczne eksperymenty. Obecnie jednym z największych wyzwań wydaje się być detekcja na obrazach, sekwencjach zdjęć, na których występuje zmienne tło. W rozdziale przedstawione zostaną współcześnie wykorzystywane metody detekcji twarzy. W przypadku niektórych z nich możliwe jest wprowadzenie modyfikacji (przez zamianę wzorca) pozwalających wykrywać inne obiekty (np. pieszych, samochody, rowery). Pierwszym algorytmem detekcji twarzy, który omówiono, będzie detekcja z użyciem koloru, odpowiadającemu kolorowi skóry. W kolejnej części omówiono algorytm Paula Viola i Michela Jonesa, a także detekcję przy użyciu algorytmu LBP oraz sieci neuronowych.

2.1. Segmentacja obszarów o kolorze skóry

Zmysł wzroku człowieka przystosowany jest do postrzegania świata poprzez analizę różnych jego cech. Może to być kolor, kształt czy dynamika ruchu. Warto wiedzieć, że człowiek postrzega barwy w pewnym zakresie widma. Przyjmuje się, że osoba zdrowa jest w stanie rejestrować długości fali w zakresie 380-780 nm. Kolor obiektu może stanowić podstawową informację, która powinna wywołać natychmiastową reakcję. Dobrym przykładem jest kolor czerwony, który powinien sygnalizować niebezpieczeństwo. Segmentacja koloru jest stosunkowo prostą operacją, która może nieść ze sobą dużą ilość informacji. Dlatego też jest często wykorzystywana w systemach detekcji oraz interpretacji sceny. Poniżej przedstawiono opis algorytmu wykorzystującego informację o samym kolorze.

Ważną cechą algorytmów opartych na detekcji koloru skóry jest stosunkowo mała złożoność obliczeniowa. Segmentacja części obrazów o określonej barwie skutkuje szybkim działaniem oraz możliwością łatwiej implementacji w układach rekonfigurowalnych. Aby jednak w pełni wykorzystać to rozwiązanie, konieczne jest rozszerzenie systemu przez dokładną ocenę parametrów takich jak wielkość czy kształt wyodrębnionej części obrazu.

Głównym elementem algorytmów opartych na kolorze jest wykorzystanie odpowiedniej przestrzeni barw. Zwykle zastosowanie znajduje przestrzeń YCbCr, która rozdziela składową luminancji (Y) od chrominancji (Cb, Cr). Kolejnym nieodłącznym elementem jest binaryzacja obrazu, odfiltrowanie oraz oznaczenie wykrytych regionów oraz obliczenie ich parametrów (wielkość, kształt, środek ciężkości).

Dekompozycja obrazu na składową luminancji i dwóch chrominancji Cb, Cr daje korzyści wynikające ze zmniejszenia wpływu oświetlenia na działanie algorytmu. Pozwala również na wykorzystanie własności jaką jest mała zmiana wartości chrominancji dla różnych ras oraz stosunkowo mały zakres odpowiadający kolorowi skóry. Dzięki temu możliwe jest jej wykrycie przy jednoczesnej eliminacji tła. Wybór zakresu parametrów dla chrominancji musi być wystarczająco wąski, aby zmaksymalizować odrzucenie pikseli odpowiadających tłu sceny. Należy jednak uważać, aby nie zawęzić przedziału za bardzo, gdyż spowoduje to zignorowanie regionów, które chcemy wykryć. Aby poprawić skuteczność działania można rozważyć połączenie kilku transformacji z użyciem różnych przestrzeni barw, np. RGB, YCbCr, HSV.

Parametry wykorzystywane do oceny, czy dany piksel wskazuje kolor skóry, czy też nie, powinny być dobrane statystycznie na dużej próbce obrazów. Prawie na pewno obraz po segmentacji (binarny) zawierał będzie szumy w postaci pojedynczych niewielkich grup pikseli, które można wyeliminować przez filtrację obrazu np. przez zastosowanie filtracji medianowej lub morfologicznej.

Na rysunku 2.1 przedstawiono przykład segmentacji obszaru o kolorze skóry.

Rysunek 2.1: Przykład detekcji koloru skóry Jain i Learned-Miller 2010

2.2. Algorytm Paula Viola i Michela Jonesa

Omawiany w niniejszym podrozdziale algorytm jest obecnie jednym z najczęściej używanych w produktach komercyjnych. Opis algorytmu napisany przez ich twórców można znaleźć w pracy [Viola i Jones 2001]. Głównymi elementami rozwiązania są cechy Haara i algorytm uczący AdaBoost.

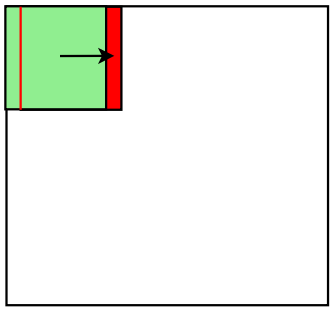
Algorytm można uznać za metodę uniwersalną, ponieważ do reprezentacji obiektu wykorzystuje proste cechy, którymi można opisać dowolny obiekt (np. samochód). Stworzenie własnego klasyfikatora wymaga jednak dostępu do dużej bazy obrazów, która będzie wykorzystywana do uczenia (setki a nawet tysiące obiektów). Proces jego uczenia może zająć nawet kilka dni na wydajnym komputerze. W dalszej części rozdziału znajduje się krótki opis działań, które należy wykonać, aby móc zacząć wykrywać wybrany przez nas obiekt. Implementacja opisywanej metody detekcji twarzy jak też innych obiektów, takich jak: oczy, nos, usta jest dostępna w oprogramowaniu *Matlab* w pakiecie *Computer Vision System Toolbox* (użycie algorytmu sprowadza się do kilkunastu linii kodu) oraz w bibliotece OpenCV. Bardziej szczegółowy opis algorytmu przedstawiono w rozdziale 6.

2.2.1. Mechanizm okna przesuwnego

Omówione w niniejszym podrozdziale zagadnienie jest wykorzystywane w algorytmach detekcji omawianych w niniejszej pracy, m.in w algorytmie Viola Jones oraz LBP (ang.*Local Binary Patterns*).

Metoda przesuwnego okna pozwala na przeskanowanie dowolnie dużego obrazu i znalezienie obszarów zawierających szukane elementy, np. twarze.

Technika ta polega na przesuwaniu okna detekcji o zdefiniowanym rozmiarze po całej ramce obrazu. Przemieszczanie okna detekcji może być realizowane co jeden bądź większą liczbę pikseli. Należy mieć na uwadze, że przesuwanie okna o pojedyncze piksele może znacząco wpłynąć na szybkość działania całego rozwiązania. Jako przykład można podać algorytm Viola Jones, który w części implementacji przesuwa okno detekcji o kilka pikseli, zmniejszając tym samym ilość potrzebnych do wykonania obliczeń. Prosty przykład działania przedstawiono na rysunku 2.2.

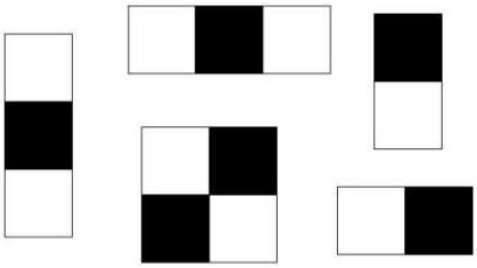


Rysunek 2.2: Mechanizm okna przesuwnego w kierunku horyzontalnym

Dodatkowym elementem o którym należy wspomnieć jest skala okna detekcji. Chcąc wykrywać obiekty o różnym rozmiarze należałoby wykorzystać technikę okna przesuwnego wielokrotnie, każdorazowo zwiększając okno o określony współczynnik – skalę.

2.2.2. Badanie własności obrazu opartych na cechach Haar-a, uczenie klasyfikatora

W poniższym rozdziale opisano cechy Haar-a oraz kroki jakie należy wykonać aby stworzyć klasyfikator wykorzystywany do detekcji. Cechy Haar-a są elementami, które w odpowiedniej ilości oraz skali są w stanie opisać niemal dowolny obiekt. Przykłady takich cech przedstawiono na rysunku. 2.3.



Rysunek 2.3: Przykład cech Haar-a wykorzystywanych w algorytmie Viola-Jones [Tomasik 2013]

Aby można było je zastosować konieczne jest przekształcenie obrazu wejściowego do skali szarości. Dla okna o rozdzielczości 24x24 pikseli występuje powyżej 160000 cech, które należałoby zbadać. Biorąc pod uwagę zbiór uczący, zawierający nieraz kilkaset obrazów pokazuje to ogrom obliczeń, jaki należy

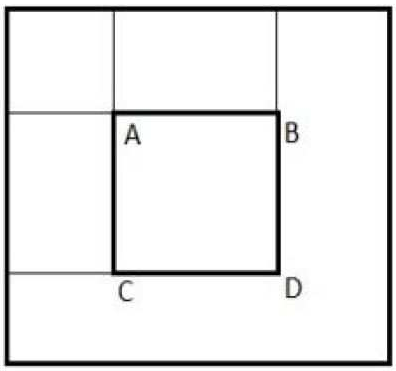
wykonać podczas uczenia i tworzenia klasyfikatora. W literaturze najpopularniejszym rozmiarem okna, dla którego tworzone są klasyfikatory jest rozmiar 20x20 pikseli, co wiąże się z większą liczbą publicznie dostępnych baz, które można wykorzystać w procesie tworzenia własnego wzorca. Do stworzenia klasyfikatora wykorzystuje się algorytm uczący AdaBoost, którego zadaniem jest wykrycie tych cech, które są w stanie odrzucić wszystkie obrazy na których nie występuje wykrywany obiekt. W przypadku klasyfikatora do detekcji twarzy dostępnego w bibliotece OpenCV twarz może zostać wykryta po sprawdzeniu 2135 cech.

2.2.3. Integral Image

Aby przyspieszyć działanie algorytmu posłużono się tzw. obrazami całkowymi (ang. *integral image*). Umożliwiają one w prosty sposób obliczyć cechy Haar'a. Element o współrzędnych (x, y) wyznaczonego obrazu ma wartość sumy wartości wszystkich pikseli leżących powyżej i na lewo od punktu, co zostało pokazane we wzorze (2.1).

$$I(x,y) = \sum_{x'<=x, y'<=y} Im(x', y') \tag{2.1}$$

$$SUM = D - B - C + A \tag{2.2}$$



Rysunek 2.4: Przykład obszaru, dla którego zastosowany może zostać wzór 2.2

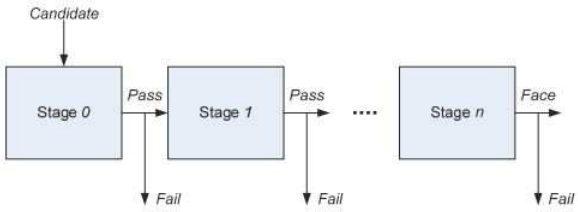
Z użyciem powyższego przekształcenia oraz mając podane współrzędne czterech punktów uzyskujemy możliwość obliczenia sumy wartości pikseli dowolnego obszaru wzorem (2.2) – tj. określonego fragmentu cechy Haar'a.

2.2.4. Detekcja obiektu przy użyciu algorytmu Viola Jones

W podrozdziale przedstawiono kroki jakie są realizowane podczas detekcji obiektu przy użyciu algorytmu Viola Jones oraz przedstawiono zalety i wady tego rozwiązania.

Detekcja przeprowadzana jest dla kolejnych okien detekcji, w których sprawdzane są cechy zapisane w klasyfikatorze. Do generacji okien może zostać wykorzystana technika przesuwnego okna opisana w podrozdziale 2.2.1. Ocena obrazu wykonywana jest w kilku etapach, co zostało przedstawione na rysunku 2.5. Klasyfikator zbudowany jest w taki sposób, że liczba cech do zbadania zwiększa się z etapu

na etap. Dzięki takiemu podejściu możliwe jest szybkie odrzucanie obrazów na których na pewno nie znajduje się szukany obiekt. Wraz z kolejnymi etapami sprawdzana jest coraz większa liczba cech, która jednoznacznie określa, czy na badanym oknie znajduje się szukany element.



Rysunek 2.5: Etapy detekcji cech [Cho i Kastner 2009]

Głównymi zaletami opisywanego algorytmu są:

- możliwość wykrywania dowolnego obiektu,
- szybkość działania,
- dostępność darmowych klasyfikatorów (np. w bibliotece OpenCV),
- wysoka skuteczność.

Niestety algorytm posiada również wady takie jak:

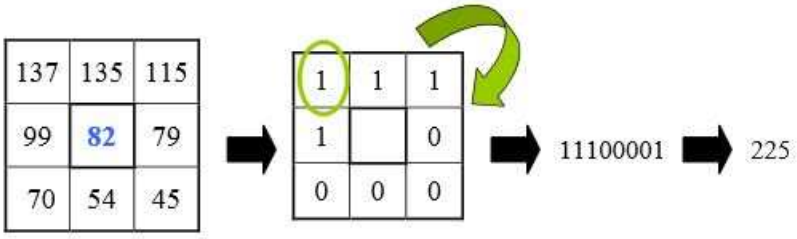
- konieczność stworzenie rozległej bazy zdjęć wykrywanego obiektu o określonej rozdzielczości przy tworzeniu nowego wzorca,
- czasochłonność procesu tworzenia klasyfikatora, wynikająca z dużej liczby obliczeń,
- brak odporności na zmianę orientacji wykrywanego obiektu (klasyfikator rozpoznający twarz na wprost, nie jest w stanie rozpoznać twarzy przedstawionej z profilu),
- wielokrotna detekcja tego samego obszaru.

2.3. LBP – Local Binary Pattern

LBP jest szeroko wykorzystywanym algorytmem bazującym na informacji zawartej w teksturze sceny, głównie z powodu jego małej złożoności obliczeniowej. Dodatkową zaletą jest możliwość jego zastosowanie zarówno do detekcji i rozpoznawania twarzy, jak też do odczytywania wyrażanych emocji. Jest również odporny na zakłócenia oświetlenia. Bardzo szczegółowy opis algorytmu wraz z otrzymanymi wynikami detekcji twarzy można znaleźć w pracy [Laura Sanchez Lopez 2010], a ogólny schemat działania wraz z dużą liczbą tytułów publikacji, w których został wykorzystany przedstawiono na stronie internetowej [*Local Binary Pattern* 2010]. Algorytm ten ma cechy wspólne z opisanym w poprzednim podrozdziale algorytmem Viola Jones-a, m.in. do opisu obrazu wykorzystuje obraz w skali szarości, technikę przesuwnego okna oraz konieczność posiadania dużych zbiorów obrazów zawierających wykrywany obiekt niezbędnych do stworzenia klasyfikatora.Głównymi elementami LBP, które umożliwiają porównywanie i znajdowanie określonych wzorców, są wektory cech. W dalszej części przedstawiono sposób ich konstruowania.

W publikacji [Laura Sanchez Lopez 2010] pokazano przykładowe cechy dla maski o rozmiarze 3x3 piksele. Obliczanie wartości liczbowej polega na porównaniu sąsiadów z pikselem centralnym, a następnie zapisaniu wyników porównania w formie bitów ułożonych w wektor. Zasadę wyznaczania wartości cechy pokazano na rysunku 2.6. Bity zapisuje się zgodnie z ruchem wskazówek zegara, zaczynając od lewego górnego rogu. Gdy wykorzystywany jest algorytm LBP często pojawia się notacja (P,R), która informuje o promieniu (R) z jakiego piksele były wybierane oraz (P) informujące o liczbie pikseli, które były porównywane. W podejściu LBP pojawianie się poszczególnych wektorów na obrazie zapisywane jest w postaci histogramu, który jest odzwierciedleniem liczy znalezionych na obrazie cech. Następnie tak zebrane dane są w dalszym ciągu porównywane ze wzorcem, a wynik porównania decyduje o sukcesie bądź porażce detekcji.

W wyniku dużego zainteresowania algorytmem pojawiały się jego modyfikacje, które znacząco przyspieszyły pracę algorytmu, nieznacznie zmniejszając przy tym jego skuteczność. Dobrym przykładem może być podzielenie cech wg. liczby zmian w wektorze. Krok ten był możliwy po zauważeniu, że pewna liczba cech powtarza się zdecydowanie częściej od pozostałych. Cechy dominujące nazywane są (ang.) *uniform patterns* i charakteryzują się małą liczbą zmian (maksymalnie 2) jaka występuje w obliczonym wektorze. Przykładem *uniform pattern* może być wektor przedstawiony na rysunku 2.6. Przykładem, w którym występuje większa liczba zmian, jest natomiast wektor: 10110101 – 6 zmian. Rozszerzenie to pozwala na znaczną redukcję badanych cech. W przypadku użycia LBP o parametrach (8, R) występuje aż 256 unikatowych cech, z czego tylko 59 wchodzi w skład *uniform pattern*. W przeprowadzonych badaniach na obrazach łączna liczba cech dominujących wahała się w przedziale od 70-90 %.

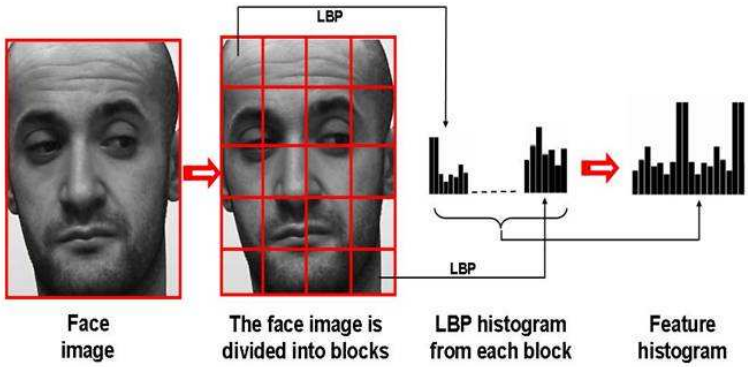


Rysunek 2.6: Przykład generacji cech LBP dla otoczenia 3x3

Aby poprawić skuteczność detekcji twarzy, obszar na którym wykrywana jest twarz dzieli się na kilka regionów, które tworzą niezależne deskryptory LBP, co pozwala wykorzystać informację o lokalizacji poszczególnych partii twarzy. Tak zebrane dane łączy się w całość i porównuje z wczesniej utworzonym wzorcem. Proces dzielenia obrazu na fragmenty przedstawiono na rysunku 2.7.

2.4. Sieci neuronowe

Wykorzystanie sztucznych sieci neuronowych jest szczególnie rozpowszechnione w sytuacjach, w których nie ma z góry ustalonych prostych reguł klasyfikacji. Sieci neuronowe, podobnie jak człowiek, mają zdolność uczenia, a zdobyta w ten sposób wiedza wykorzystywana jest do późniejszego analizowania danych. Sama budowa sieci jest zainspirowana sposobem działania mózgu, a jej głównym budulcem są neurony i połączenia pomiędzy nimi – synapsy, które to mogą przewodzić daną informację lepiej bądź gorzej. Ta zdolność przewodzenia – w postaci wagi dla każdego połączenia – podlega procesowi uczenia.



Rysunek 2.7: Podział obrazu na obszary, dla których obliczane są wartości cech w metodzie LBP [*Local Binary Pattern* 2010]

Głównymi zaletami sieci neuronowych są:

- zdolność do uogólniania danych,
- odporność na błędne dane,
- możliwość zastosowania, gdy typowe rozwiązania programowe są bardzo skomplikowane, a ich realizacja kosztowna,
- dostosowywanie się do zmiennego otoczenia.

Sztuczne sieci neuronowe nie mają ściśle z góry określonych reguł według których muszą zostać zaimplementowane. Liczba neuronów oraz warstw, a także sposób uczenia muszą zostać dobrane do konkretnego zadania detekcji, często o jej poprawności autorzy dowiadują się dopiero po szeregu eksperymentów przeprowadzonych po procesie uczenia i testach.

W pracy AL-Allaf 2014 przedstawiono i porównano 12 różnych implementacji sztucznych sieci neuronowych stworzonych do detekcji twarzy. Najlepszy wskaźnik detekcji – 97,6% osiągnęła konwolucyjna sieć neuronowa.

2.5. Przyszłość algorytmów detekcji twarzy

Opisywane w niniejszej pracy algorytmy zostały stworzone do detekcji konkretnego wzorca, np. twarzy widocznej z przodu. Niestety w przypadku zmiany kąta, bądź widoku z profilu stają się one niewystarczające i należałoby stworzyć kolejne detektory odpowiadające poszczególnym przypadkom, które chcemy wykryć. Obiecującym sposobem detekcji wydają się być konwolucyjne sieci neuronowe o dużej liczbie warstw (ang. *Deep Convolutional Neural Networks*). Przykład zastosowania takiej sieci przedstawiono w pracy [Sachin Sudhakar Farfade 2015].

Jednym z głównych wyzwań związanych z tworzeniem sieci jest ich uczenie. W wielowarstwowych sieciach neuronowych używa się rozległych baz danych (obrazy w bazach zawierają twarze przedstawione pod różnymi kątami). Do stworzenia poprawnie działającej sieci neuronowej potrzeba setek tysięcy przykładów uczących (tzw. próbek). W pracy [Sachin Sudhakar Farfade 2015] autorzy wykorzystali 200 tyś. obrazów zawierających twarze oraz 20 mln obrazów bez twarzy. Wynikiem ich prac jest algorytm

Rysunek 2.8: Poprawna detekcja twarzy pod różnym kątem z użyciem sieci neuronowej [Sachin Sudhakar Farfade 2015].

poprawnie wykrywający twarze pod różnym kątem oraz te częściowo zasłonięte. Możliwości systemu przedstawiono na rysunku 2.8.

Na uwagę zasługuje fakt, że głębokie sieci neuronowe zawierające wiele warstw coraz częściej postrzegane są jako element, który może zbliżyć naukę do stworzenia systemu wizyjnego będącego w stanie rozpoznawać obiekty z podobną dokładnością do człowieka. Technika *Deep learning* zaczęła rozwijać się stosunkowo niedawno, efektywne metody uczenia sieci składających się z wielu warstw zaczęły powstawać w 2006 r, a wraz ze wzrostem mocy obliczeniowej układów GPU (ang. *Graphics processing unit*) ich uczenie przebiega coraz szybciej. W niedalekiej przyszłości należy spodziewać się wielu usprawnień i nowych zastosowań, w tym, w systemach wizyjnych czasu rzeczywistego. W kontekście tematu niniejszej pracy warto odnotować, że jednym z kolejnych etapów *Deep Learning*-u może być coraz szersze wykorzystywanie układów FPGA do projektowania opisywanych sieci. Wskazywać na to może coraz większe zainteresowanie ze strony producentów układów FPGA oraz wprowadzanie ułatwień dla osób tworzących takie sieci. Przykładem jest rozwiązanie firmy Altera, które umożliwia stworzenie architektury sieci neuronowej przy użyciu dostarczanego SDK zintegrowanego z OpenCL [*Efficient Implementation of Neural Network Systemms Built on FPGAs, Programmed with OpenCl* 2015].

Detekcja twarzy jest gałęzią dynamicznie rozwijającą się, czekającą na pojawienie się nowych, nieszablonowych rozwiązań. Z pewnością jest i pozostanie jednym z głównym zadań systemów wizyjnych.