



**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**

**WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI**

KATEDRA Informatyki

## **PROJEKT INŻYNIERSKI**

**Rozpoznawanie ekspresji twarzy w oparciu o sieci neuronowe  
konwolucyjne**

Dokumentacja procesowa

Autorzy:  
Kierunek studiów:  
Opiekun pracy:

Jacek Oblaza, Dmytro Petruk  
Informatyka  
Dr hab. Bogdan Kwolek

Kraków, 2016



## 1. Wizja

Celem projektu jest stworzenie biblioteki, która będzie potrafiła wykrywać ekspresję twarzy na dostarczonym obrazie. Biblioteka ta będzie w stanie przypisać do danej twarzy na zdjęciu jedną z następujących ekspresji: złość, pogarda, zniesmaczenie, strach, radość, smutek, zaskoczenie. Będzie można ją zastosować w komunikatorach i czatach internetowych, aby poprzez pobranie obrazu twarzy z kamery, program odczytał ekspresję i zamienił ją na odpowiadającą emotikonę.

## 2. Wymagania

### Funkcjonalne:

- Rozpoznawanie ekspresji twarzy na podstawie dostarczonego obrazu
- Możliwość poprawiania osiągnięć dla twarzy, które będą występować częściej

### Niefunkcjonalne:

- Działanie w czasie rzeczywistym
- API
- Działanie w trybie offline

## 3. Przegląd rozwiązań

Na rynku istnieją gotowe rozwiązania dla osiągnięcia celu rozpoznawania emocji, ale każde z nich ma jakieś wady.

- Microsoft Cognitive Services Emotion API [1]. Teraz jest jednym z najlepszych rozwiązań na rynku i z dostępną polityką cenową. Główną jego wadą, jak i większości innych rozwiązań jest to, że działa jako SaaS (Software as a Service), więc istnieje możliwość użycia tylko przy dostępie do sieci, co nie zawsze jest możliwe.
- Kairos [2]. Wadą jest brak możliwości użycia w trybie offline.
- EmoVu [3]. Tak jak wcześniejsze rozwiązania jedyny możliwy typ pracy to online.
- Nviso [4]. Chociaż jest uważany za bardzo dobre rozwiązanie ze strony technicznej, ale nie ma możliwości edycji, czy jakiegokolwiek zmiany.
- Nvidia DIGITS + DetectNet [5], chociaż nie służy do rozpoznawania emocji, a do rozpoznawania obiektów, ale zasadniczo działa podobnie do naszej biblioteki. Pozwala na podanie zbioru treningowego i uruchomienia uczenia sieci, przy jednoczesnym braku konieczności przejmowania się szczegółami technicznymi.

Jak widać żadne rozwiązanie nie spełnia wcześniejszych wymagań, ponieważ albo działają tylko online, albo nie mają możliwości poprawy osiągnięć, dla danych twarzy. Ostatnie rozwiązanie jest najbliższe spełnieniu wymagań, jednak nie dostarcza gotowej sieci stworzonej do wykrywania ekspresji, tylko trzeba samemu zebrać zbiór treningowy i wytrenować sieć. My chcielibyśmy jednak, aby nasze rozwiązanie do poprawnego działania nie wymagało od użytkownika niczego więcej niż tylko instalacji.

## 4. Metodyka

Jako Metodologię projektowania systemu wybraliśmy połączenie Agile [6] i Lean Programming [7], co pozwoliło na działanie iteratywne, inkrementalne i ewolucyjne w rozbudowie produktu wraz z krótkimi cyklami dla eliminacji działań nie przyczyniających się do realizacji wyznaczonych celów.

## 5. Narzędzia

Jako elementy Agile wykorzystano Kanban board [8] (trello [9]) i pair programming [10]. Jako systemu kontroli wersji użyto git[11] z na repozytorium na GitHub [12]. Są to bardzo popularne rozwiązania, a także obaj korzystaliśmy z nich wcześniej i wiemy, że spełnią swoje zadania.

## 6. Koncepcja rozwiązania

Najsensowniejszym wydaje się nam użyć najnowszych osiągnięć w dziedzinie uczenia głębokiego jakimi są sieci neuronowe konwolucyjne. Idealnie nadają się one do rozpoznawania obrazów i wykrywania wspólnych cech obrazów, takich jak mimika twarzy. Sieć taka będzie wymagała dużego zbioru treningowego oraz czasu na naukę.

Wejściem do sieci będzie zdjęcie danej osoby. Aby nie było za dużo niepotrzebnych danych (pikseli) potrzebna nam sama twarz. Co za tym idzie przed podaniem danego zdjęcia na wejście należy wykryć na nim twarz i przeskalować obszar z samą twarzą do wybranego przez nas rozmiaru. Tu rodzi się kolejny problem wykrywania twarzy na obrazach. Wyjściem sieci będą stopnie nasycenia twarzy daną ekspresją.

## 7. Podział prac

Dmytro Petruk	Jacek Obłaza
- Przeprowadzenie analizy dostępnych narzędzi	- Implementacja wykrywania twarzy
- Przygotowanie zbioru treningowego	i dostosowywania jej do wejścia sieci
- Implementacja wczytywania danych	- Implementacja trenowania sieci neuronowej
- Trenowanie i ewaluacja sieci	- Trenowanie i ewaluacja sieci

## 8. Analiza ryzyka

Ryzyko	Poziom zagrożenia (1 - 10)	Przeciwdziałanie	Działanie w razie wystąpienia
Błędne nauczanie sieci neuronowej (w tym przeuczenie)	7	Wielokrotne uczenie sieci z różnymi parametrami i wybór najlepszych na podstawie uzyskanych wyników	Ponowne uczenie przy zmienionych parametrach lub zbiorze treningowym. Zmniejszenie ilości rozpoznawanych ekspresji

Niewystarczający rozmiar istniejących zbiorów danych	5	Znalezienie odpowiednio dużego zbioru danych (co najmniej 2000 zdjęć z podpisami)	Rozszerzenie ich o własny zbiór danych
Niedopuszczalnie długi czas rozpoznawania	3	Testowanie czasu działania przy różnych rozmiarach wejścia sieci i różnej ilości warstw, neuronów, połączeń w architekturze sieci.	Zmniejszenie rozmiarów sieci, bądź też rozmiaru obrazu wejściowego. Ograniczenie się do obrazu samych szczególnych części twarzy, takich jak usta, oczy czy nos
Nieodpowiednie narzędzia	2	Dokładna analiza dostępnych narzędzi, w tym przetestowanie wybranych	Zmiana narzędzi

[1] <https://www.microsoft.com/cognitive-services/en-us/emotion-api>

[2] <https://www.kairos.com/>

[3] <http://emovu.com/>

[4] <http://www.nviso.ch/>

[5] <https://developer.nvidia.com/digits>

[6] [https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development)

[7] [https://en.wikipedia.org/wiki/Lean\\_software\\_development](https://en.wikipedia.org/wiki/Lean_software_development)

[8] [https://en.wikipedia.org/wiki/Kanban\\_board](https://en.wikipedia.org/wiki/Kanban_board)

[9] <https://trello.com/>

[10] [https://en.wikipedia.org/wiki/Pair\\_programming](https://en.wikipedia.org/wiki/Pair_programming)

[11] <https://en.wikipedia.org/wiki/Git>

[12] <https://github.com>