

**SYLABUS**  
**DOTYCZY CYKLU KSZTAŁCENIA 2022-2024**  
Rok akademicki 2023/2024

**1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE**

Nazwa przedmiotu	<i>analiza i przetwarzanie obrazów</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>Informatyka</i>
Poziom studiów	<i>studia II stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok II, semestr 2</i>
Rodzaj przedmiotu	<i>przedmiot kształcenia kierunkowego</i>
Język wykładowy	<i>polski</i>
Koordynator	<i>dr inż. Michał Kępski</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Michał Kępski</i>

**1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS**

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
2	15			15					3

**1.2. Sposób realizacji zajęć**

zajęcia w formie tradycyjnej

**1.3 Forma zaliczenia przedmiotu (z toku)**

zaliczenie z oceną

**2. WYMAGANIA WSTĘPNE**

Znajomość podstaw algebry i analizy matematycznej. Znajomość podstawowych algorytmów i struktur danych. Podstawowa umiejętność programowania w środowisku Python.

### 3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

#### 3.1 Cele przedmiotu

C <sub>1</sub>	Poznanie wybranych metod przetwarzania i analizy obrazów.
C <sub>2</sub>	Zapoznanie z tematyką zagadnienia widzenia komputerowego i powiązań z pokrewnymi zagadnieniami (uczenie maszynowe, sztuczna inteligencja). Zapoznanie studentów z typowymi problemami i wyzwaniem stawianymi przed algorytmami/systemami przetwarzającymi/analizującymi obrazy cyfrowe.
C <sub>3</sub>	Poznanie środowiska Python NumPy i biblioteki OpenCV, nabycie umiejętności wykorzystywania ich we własnych projektach informatycznych.

#### 3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Student zna wybrane metody przetwarzania obrazu, ich podstawy matematyczne oraz klasy problemów, do których metody te są wykorzystywane.	K_Wo1, K_Wo3
EK_02	Student potrafi wyszczególnić metody/grupy metod widzenia komputerowego będące <i>state-of-the-art</i> w zależności od zadanego problemu.	K_Wo4
EK_03	Student zna sposoby implementacji metod przetwarzania obrazów, jest świadomy ich wymagań obliczeniowych i złożoności, potrafi wskazać architektury sprzętowe i techniki programowania wykorzystywane przy ich implementacji.	K_Wo3, K_Wo6
EK_04	Student potrafi stosować algorytmy przetwarzania i analizy obrazów oraz interpretować wyniki ich działania.	K_Uo4, K_Uo6
EK_05	Student potrafi parametryzować algorytmy przetwarzania i analizy obrazów w zależności od rozwiązywanego problemu.	K_Uo5, K_Uo6

#### 3.3 Treści programowe

##### A. Problematyka wykładu

1. Podstawowe zagadnienia widzenia komputerowego. Sposoby akwizycji i reprezentacji obrazów kolorowych i RGB-D.
2. Operacje bezkontekstowe. Konwolucje, filtry nieliniowe. Gradient i detekcja krawędzi.
3. Transformaty: Fouriera, DCT, Hougha i ich zastosowanie w przetwarzaniu obrazów.
4. Detekcja narożników. Detekcja punktów charakterystycznych. SIFT. Deskryptory cech.
5. Model i kalibracja kamery. <i>Feature based alignment</i> . Algorytm RANSAC. <i>Multiple view geometry</i> .
6. Rozpoznawanie obiektów. K-NN, klasyfikacja liniowa, SVM. <i>Stochastic gradient descent</i> .
7. Konwolucyjne sieci neuronowe. Wielowarstwowy perceptron i algorytm wstecznej propagacji. Uczenie głębokie. Warstwy konwolucyjne i <i>pooling</i> . Zastosowania, <i>state-of-the-art</i> .
8. Metody przetwarzania chmur punktów 3D. Urządzenia do akwizycji. Segmentacja, filtracja, algorytm ICP.

## B. Problematyka ćwiczeń laboratoryjnych

1. Biblioteka OpenCV i numpy. Podstawowe operacje na obrazach. Jupyter notebook jako środowisko do prezentacji wyników.
2. Filtracja obrazu. Wykrywanie krawędzi.
3. Detekcja narożników. Detekcja punktów charakterystycznych. SIFT.
4. Przepływ optyczny (ang. <i>optical flow</i> ).
5. Klasyfikacja obiektów. Proces uczenia klasyfikatora liniowego.
6. Konwolucyjne sieci neuronowe: biblioteki programistyczne, proces uczenia prostej sieci do rozpoznawania obrazów (zbiór CIFAR-10).
7. Metody przetwarzania chmur punktów 3D (biblioteka Open3D)
8. Model i kalibracja kamery. Algorytm RANSAC. <i>Multiple view geometry</i> .

### 3.4 Metody dydaktyczne

Wykład: prezentacja multimedialna

Laboratorium: implementacja i testowanie algorytmów przetwarzania i analizy obrazów.

## 4. METODY I KRYTERIA OCENY

### 4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych
EK_01	Test zaliczeniowy	wykład
EK_02	Test zaliczeniowy	wykład
EK_03	Test zaliczeniowy	wykład
EK_04	Kolokwium, obserwacja w trakcie zajęć	lab
EK_05	Kolokwium, obserwacja w trakcie zajęć	lab

### 4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Zaliczenie wykładu – test – efekty EK\_01, EK\_02, EK\_03

91 – 100% bardzo dobry (5.0); 81 – 90% plus dobry (4.5); 71 – 80% dobry (4.0);

61 – 70% plus dostateczny (3.5); 50 – 60% dostateczny (3.0); poniżej 50% niedostateczny (2.0).

Zaliczenie laboratorium – kolokwium – efekty EK\_04, EK\_05

DST: Student potrafi poprawnie przygotować programy (uzupełniając udostępniony szkielet kodu źródłowego) rozwiązujące proste problemy, wykorzystując do tego celu możliwości bibliotek programistycznych używanych na zajęciach. Potrafi parametryzować metody przetwarzania obrazów zgodnie z zadanymi wskazówkami.

DB: Student potrafi poprawnie przygotować programy (uzupełniając udostępniony szkielet kodu źródłowego) rozwiązujące średniozaawansowane problemy, wykorzystując do tego celu możliwości bibliotek programistycznych używanych na zajęciach. Zna podstawowe parametry

danych metod i wpływ ich modyfikacji na działanie algorytmów. Potrafi parametryzować metody przetwarzania obrazów zgodnie z zadanymi wskazówkami.

BDB: Student potrafi poprawnie przygotować programy (uzupełniając udostępniony szkielet kodu źródłowego), rozwiązujące średniozaawansowane problemy wykorzystując do tego celu możliwości bibliotek programistycznych używanych na zajęciach. Zna parametry danych metod i wpływ ich modyfikacji na działanie algorytmów. Potrafi zinterpretować poprawnie wyniki i samodzielnie parametryzować wykorzystywane metody w zależności od zadanego problemu.

Ocena końcowa może być podwyższona lub obniżona (o pół stopnia) na podstawie aktywności studenta na zajęciach lub jej braku.

#### 5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	30
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	45
SUMA GODZIN	75
<b>SUMARYCZNA LICZBA PUNKTÓW ECTS</b>	<b>3</b>

#### 6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

#### 7. LITERATURA

Literatura podstawowa:

1. Szeliski R.: *Computer vision: Algorithms and applications*, 2nd edition, Springer, 2022. (dostępna w internecie)
2. Klette R.: *Concise Computer Vision: An Introduction into Theory and Algorithms. Undergraduate Topics in Computer Science*, Springer-Verlag London, 2014.

Literatura uzupełniająca:

1. Cyganek B., *An Introduction to 3D Computer Vision Techniques and Algorithms*, Wiley, 2009.
2. McKinney W.: *Python w analizie danych: przetwarzanie danych za pomocą pakietów Pandas i NumPy oraz środowiska IPython*, Helion, 2018.