

Mateusz SZWEDKA, Sebastian BUDZAN
Politechnika Śląska

ANALIZA OGRANICZEŃ MODELU FCN_RESNET50 WZGLĘDEM ARCHITEKTUR TRANSFORMEROWYCH W ZADANIU DETEKCJI PRZESZKÓD NA POTRZEBY PLANOWANIA TRAJEKTORII

Streszczenie. FCN_ResNet50 może poprawić bezpieczeństwo nawigacji względem klasycznego A*, ale w praktyce wykazuje tendencję do nadmiernie konserwatywnego zachowania, co wydłuża misję i zwiększa liczbę korekt ruchu. Architektura transformerowa może ograniczać te problemy dzięki stabilniejszej masce segmentacyjnej. W artykule postarano się przedstawić te różnice, gdzie w tym celu wykorzystano środowisko Gazebo działające w ramach ROS2. Zaimplementowany A* dostał wspomaganie zarówno FCN_ResNet50 oraz vTransformerem SegFormer-B0.

ANALYSIS OF THE LIMITATIONS OF FCN_RESNET50 COMPARED TO TRANSFORMER ARCHITECTURES IN OBSTACLE DETECTION FOR TRAJECTORY PLANNING

Summary. FCN_ResNet50 can improve navigation safety compared to the classic A* algorithm, but in practice it tends to behave too conservatively, which prolongs the mission and increases the number of motion corrections. The transformer architecture may mitigate these issues thanks to a more stable segmentation mask. This paper aims to highlight these differences, using the Gazebo environment running under ROS2. The implemented A* algorithm was augmented with both FCN_ResNet50 and the vTransformer SegFormer-B0.

1. Wstęp

Planowanie ścieżki (ang. *path planning*) jest jednym z fundamentalnych zagadnień, na których oparte są badania nad autonomicznością pojazdów, takich jak roboty mobilne typu AMR lub AGV. W rzeczywistych zastosowaniach pojazd autonomiczny powinien nie tylko wyznaczyć trasę prowadzącą do celu, ale przede wszystkim unikać przeszkód, ograniczać liczbę manewrów korekcyjnych, zachowywać płynność ruchu i unikać niepotrzebnych zatrzymań. Problem ten jest szczególnie istotny w środowiskach wewnętrznych, takich jak korytarze, pomieszczenia mieszkalne lub hale przemysłowe, gdzie taki robot porusza się często w pobliżu ścian, mebli oraz innych obiektów. W literaturze często stosowane są klasyczne algorytmy planowania, takie jak algorytm zupełny Dijkstra lub heurystyczny A*, a także ich liczne modyfikacje [1, 4, 10].

Algorytm A* jest popularnym rozwiązaniem ze względu na stosunkowo prostą implementację, deterministyczny charakter oraz możliwość efektywnego przeszukiwa-

nia przestrzeni z wykorzystaniem funkcji heurystycznej, która może dawać satysfakcjonujące efekty. Jego skuteczność zależy jednak w krytycznym stopniu od jakości mapy środowiska. W przypadku pojazdów mobilnych mapa taka jest często budowana z wykorzystaniem różnych algorytmów percepcyjnych (np. SLAM - Simultaneous Localization and Mapping), których niedoskonałością może być ich podatność na błędy wynikające z ograniczeń sensorów, niepełnej obserwacji otoczenia lub pojawienia się przeszkód nieuwzględnionych wcześniej w mapie [5, 7, 8]. Z tego względu proponowane są podejścia hybrydowe, w których klasyczne algorytmy planowania są uzupełniane o dodatkowe informacje pochodzące z różnych systemów percepcyjnych.

Jednym z takich sposobów zwiększenia bezpieczeństwa nawigacji pojazdu autonomicznego jest wykorzystanie segmentacji obrazu do wyznaczania obszarów potencjalnie zajętych przez przeszkody. W poprzednich badaniach autorów zastosowano podejście łączące algorytm A^* z siecią FCN_ResNet50 [6], co pozwoliło wzbogacić mapę zajętości o informacje pochodzące z kamery [9]. Rozwiązanie to może jednak prowadzić do zbyt konserwatywnego (ostrożnego) zachowania robota, szczególnie w momentach, w których maska segmentacyjna traktuje zbyt duży obszar ROI na obrazie jako przeszkodę. W konsekwencji robot może wykonywać dodatkowe niepotrzebne z punktu widzenia płynności przejazdu manewry korekcyjne, częściej zatrzymywać się lub realizować misję w dłuższym czasie.

Celem niniejszego artykułu jest porównanie klasycznego algorytmu A^* oraz dwóch wariantów wspomaganych percepcją wizyjną: A^* z siecią FCN_ResNet50 oraz A^* z lekkim modelem transformerowym typu SegFormer-B0. Szczególną uwagę zwrócono na ograniczenia praktycznego wykorzystania FCN_ResNet50 w zadaniu detekcji przeszkód oraz na możliwość ograniczenia tych problemów przez zastosowanie architektury transformerowej, która lepiej wykorzystuje globalny kontekst sceny [2, 3, 12]. W artykule przedstawiono środowisko testowe, zastosowane metody oraz porównanie wyników uzyskanych dla trzech wariantów planowania.

2. Opis problemu i zastosowane metody

2.1. Założenia porównania

Celem pracy jest porównanie działania dwóch podejść do percepcji wizyjnej wspomagającej planowanie ścieżki robota mobilnego. W obu przypadkach wykorzystano ten sam algorytm planowania globalnego A^* oraz tę samą mapę zajętości środowiska. Różnica pomiędzy badanymi wariantami dotyczy sposobu wyznaczania maski przeszkód na podstawie obrazu z kamery. W pierwszym przypadku zastosowano klasyczną sieć konwolucyjną FCN_ResNet50, natomiast w drugim lekki model typu vision-transformer SegFormer-B0 [11].

Klasyczny algorytm A^* wykorzystano jako wspólną podstawę systemu planowania. Pozwala to ograniczyć wpływ innych czynników i skoncentrować analizę na jakości informacji dostarczanej przez moduł percepcji wizyjnej. W badaniach uwzględniono również wariant bazowy, w którym robot korzysta wyłącznie z mapy zajętości generowanej przez system RTAB-Map. Wariant ten stanowi punkt odniesienia dla metod wykorzystujących segmentację obrazu.

W pracy nie analizowano szczegółowo modyfikacji samego algorytmu planowania. Istotą porównania jest praktyczny wpływ zastosowanego modelu segmentacyjnego

na zachowanie robota, w szczególności na bezpieczeństwo przejazdu, płynność ruchu, liczbę zatrzymań, liczbę skrętów oraz czas realizacji misji.

2.2. Wariant z siecią FCN_ResNet50

FCN_ResNet50 jest przykładem w pełni konwolucyjnej sieci segmentacyjnej, w której ekstraktor cech oparty jest na architekturze ResNet50. Sieci tego typu są często wykorzystywane w zadaniach segmentacji semantycznej, ponieważ umożliwiają przypisanie etykiety do każdego piksela obrazu. Ich zaletą jest stosunkowo prosta implementacja oraz dostępność gotowych modeli w popularnych bibliotekach uczenia głębokiego. W analizowanym systemie sieć FCN_ResNet50 przetwarza obraz z kamery i generuje maskę obszarów interpretowanych jako przeszkody. Maskę tę jest następnie wykorzystywana przez system planowania jako dodatkowa informacja o otoczeniu robota. Dzięki temu robot może reagować na przeszkody widoczne w obrazie z kamery, nawet jeżeli nie zostały one wystarczająco dokładnie odwzorowane na mapie SLAM [9].

Praktycznym ograniczeniem tego rozwiązania jest jednak tendencja do nadmiernej konserwatywnej interpretacji sceny. W przypadku gdy model traktuje zbyt duży fragment obrazu jako potencjalną przeszkodę, robot może wykonywać niepotrzebne manewry korekcyjne lub omijać obszary, które w rzeczywistości nie stanowią zagrożenia. Może to prowadzić do zwiększenia czasu przejazdu, większej liczby lokalnych korekt oraz mniej płynnej trajektorii.

2.3. Wariant z modelem transformerowym

Drugim badanym podejściem jest zastosowanie lekkiego modelu typu vision-transformer SegFormer-B0. Modele tego typu wykorzystują mechanizm uwagi, który pozwala uwzględniać szerszy kontekst sceny niż klasyczne sieci konwolucyjne [2, 3]. W zadaniu segmentacji przeszkód może to mieć istotne znaczenie, ponieważ poprawna interpretacja obrazu często wymaga rozróżnienia obszarów podłogi, ścian, mebli i innych obiektów występujących w otoczeniu robota. W badanym systemie model SegFormer-B0 pełni taką samą rolę jak FCN_ResNet50, czyli generuje maskę przeszkód na podstawie obrazu z kamery. Dzięki temu możliwe jest bezpośrednie porównanie obu podejść w tych samych warunkach eksperymentalnych. Różnice w zachowaniu robota wynikają więc przede wszystkim z jakości i stabilności masek generowanych przez oba modele.

Zastosowanie modelu transformerowego może ograniczać problem fałszywych lub nadmiernie rozległych detekcji, ponieważ model lepiej uwzględnia relacje pomiędzy różnymi fragmentami obrazu. W konsekwencji robot powinien wykonywać mniej zbędnych korekt, poruszać się płynniej i lepiej utrzymywać wyznaczoną trajektorię. W dalszej części pracy przedstawiono wyniki porównania obu wariantów w środowisku symulacyjnym.

3. Środowisko testowe

Badania przeprowadzono w środowisku symulacyjnym Gazebo, pracującym w ramach ROS 2. Symulacja obejmowała robota mobilnego poruszającego się w środowisku wewnętrznym, zawierającym korytarze, wąskie przejścia, ściany oraz przeszkody znajdujące się w pobliżu planowanej trasy. Taki układ środowiska pozwala sprawdzić, czy zastosowanie dodatkowej percepcji wizyjnej wpływa na bezpieczeństwo i płynność

ruchu robota.

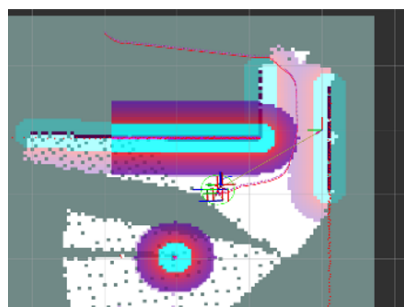
Do budowy mapy środowiska wykorzystano system RTAB-Map, generujący mapę zajętości wykorzystywaną następnie przez algorytm planowania globalnego. Robot poruszał się pomiędzy wyznaczonymi punktami pośrednimi, a dla każdego badanego wariantu analizowano przebieg rzeczywistej trajektorii, czas realizacji misji oraz występowanie kolizji. Wariant bazowy korzystał wyłącznie z mapy zajętości, natomiast warianty wizyjne wykorzystywały dodatkowo obraz z kamery przetwarzany przez model segmentacyjny.

W badaniach porównano trzy warianty systemu:

- klasyczny algorytm A^* oparty na mapie RTAB-Map,
- algorytm A^* wspomagany maską przeszkód wyznaczaną przez FCN_ResNet50,
- algorytm A^* wspomagany maską przeszkód wyznaczaną przez model transforme-
rowy SegFormer-B0.

W każdym wariancie wykorzystano tę samą mapę środowiska oraz ten sam zestaw punktów docelowych. Dzięki temu możliwe było porównanie wpływu zastosowanego modułu percepcji na zachowanie robota, bez zmiany podstawowego algorytmu planowania. Szczególną uwagę zwrócono na to, czy wariant wizyjny pozwala ograniczyć liczbę kolizji oraz czy nie powoduje jednocześnie nadmiernego wzrostu liczby manewrów korekcyjnych.

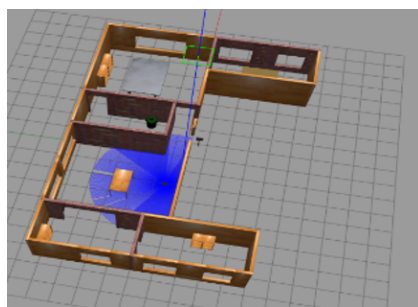
Na rysunku 1 przedstawiono przykładowy stan działania systemu testowego. Widoczna jest mapa kosztów z wyznaczoną ścieżką, wynik segmentacji obrazu, model środowiska w Gazebo oraz obraz z kamery robota. Rysunek ten ilustruje sposób, w jaki informacja wizyjna była wykorzystywana do oceny otoczenia robota podczas realizacji zadania nawigacyjnego.



(a) Mapa kosztów



(b) Maska segmentacji



(c) Model w Gazebo



(d) Widok z kamery

Rys. 1. Przykładowy stan środowiska testowego: mapa kosztów, wynik segmentacji, model symulacyjny oraz widok z kamery robota.

Do oceny działania badanych wariantów wykorzystano zestaw metryk opisujących zarówno jakość planowania, jak i zachowanie robota podczas ruchu. Uwzględniono długość ścieżki planowanej i rzeczywistej, średnie oraz maksymalne odchylenie od planu, liczbę skrętów, liczbę zatrzymań, czas manewrów korekcyjnych, liczbę kolizji, liczbę eksplorowanych węzłów oraz całkowity czas realizacji misji. Zestaw tych miar pozwala ocenić nie tylko fakt osiągnięcia celu, ale również płynność i bezpieczeństwo ruchu.

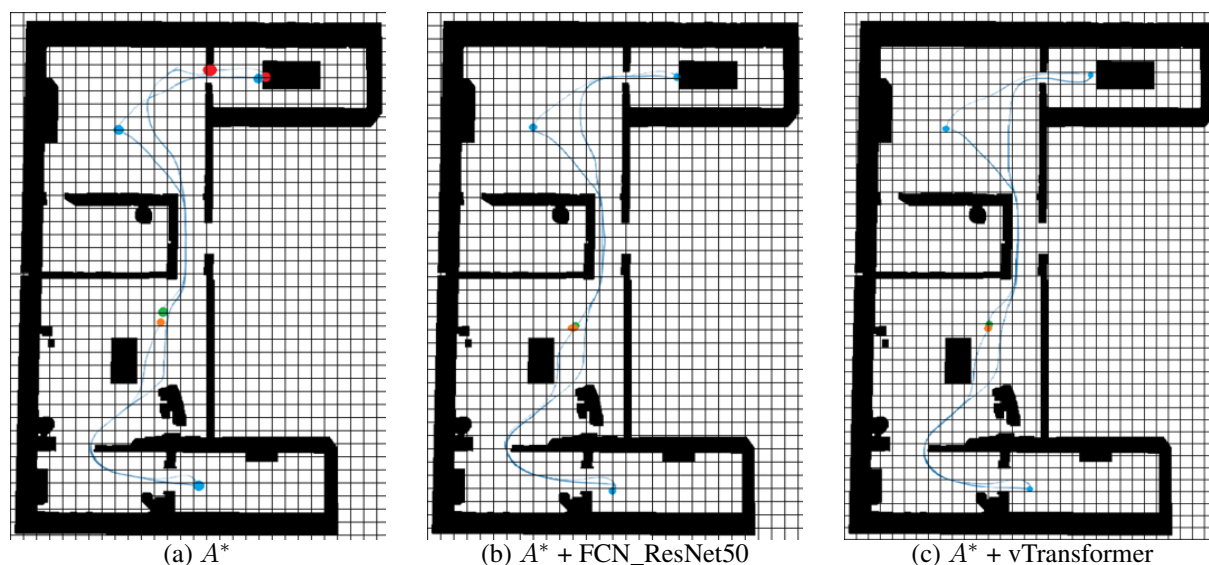
4. Wyniki badań

4.1. Porównanie przebiegu trajektorii

W pierwszym etapie porównano przebieg trajektorii uzyskanych dla trzech wariantów systemu. Wariant bazowy, wykorzystujący wyłącznie algorytm A^* oraz mapę RTAB-Map, pozwolił robotowi osiągnąć punkt docelowy, jednak przejazd odbył się z kolizjami. Oznacza to, że sama mapa zajętości nie zapewniła wystarczającej informacji o bezpiecznym przejeździe w pobliżu przeszkód.

Zastosowanie percepcji wizyjnej pozwoliło wyeliminować kolizje zarówno w przypadku wariantu z FCN_ResNet50, jak i wariantu z modelem transformerowym. Różnice pomiędzy tymi podejściami są jednak widoczne w charakterze ruchu robota. Wariant z FCN_ResNet50 wykazuje bardziej zachowawcze działanie. Robot częściej wykonuje lokalne korekty toru ruchu, co wskazuje na ostrożniejszą interpretację obszarów znajdujących się przed kamerą. Z kolei wariant z modelem transformerowym generuje bardziej płynną trajektorię, z mniejszą liczbą lokalnych odchyień i zatrzymań.

Na rysunku 2 przedstawiono porównanie przebiegu tras dla trzech badanych wariantów. Kolorem niebieskim oznaczone punkty, które były wyznaczane sekwencyjnie, tzn. po osiągnięciu jednego z nich, natychmiast wyznaczano kolejny. Zadanie pojazdu autonomicznego było osiągnięcie każdego z nich. Kolorem czerwonym natomiast wyróżniono miejsca, gdzie doszło do kolizji. Kolor zielony natomiast oznacza punkt początkowy trasy, a pomarańczowy punkt końcowy. Zauważyć można, że wariant bazowy prowadzi do niebezpiecznego przejazdu, natomiast oba warianty wizyjne pozwalają uniknąć kolizji. Najbardziej stabilny przebieg uzyskano dla wariantu z modelem transformerowym.



Rys. 2. Porównanie przebiegu trajektorii dla trzech wariantów systemu planowania.

4.2. Porównanie wyników ilościowych

Wyniki ilościowe zestawiono w tabeli 1. Uwzględniono metryki opisujące długość ścieżki, odchylenie od planu, liczbę skrętów i zatrzymań, czas manewrów korekcyjnych, liczbę kolizji, liczbę eksplorowanych węzłów oraz czas realizacji misji.

Tabela 1

Porównanie wyników dla trzech wariantów planowania

Parametr	A^*	$A^* + \text{FCN_ResNet50}$	$A^* + \text{vTransformer}$
Długość planowana [m]	35.367	37.013	35.949
Długość rzeczywista [m]	37.513	36.209	35.073
Różnica długości [m]	+2.146	-0.894	-0.876
Średnie odchylenie [m]	0.061	0.051	0.046
Maksymalne odchylenie [m]	0.194	0.259	0.131
Liczba skrętów	107	86	72
Liczba zatrzymań	9	6	5
Manewry korekcyjne [s]	12.58	31.32	16.22
Liczba kolizji	3	0	0
Osiągnięcie celu	tak, z kolizjami	tak	tak
Liczba eksplorowanych węzłów	651	682	664
Czas misji [s]	478.77	822.8	669.72

Wariant bazowy A^* uzyskał najkrótszy czas realizacji misji, równy 478.77 s, jednak wynik ten nie oznacza najlepszego działania systemu, ponieważ przejazd zakończył się trzema kolizjami. Robot poruszał się szybciej, ale kosztem bezpieczeństwa. Dodatkowo rzeczywista długość ścieżki była większa od długości planowanej, co wskazuje na odchylenia od pierwotnie wyznaczonego toru ruchu.

Wariant $A^* + \text{FCN_ResNet50}$ wyeliminował kolizje, co potwierdza użyteczność dodatkowej percepcji wizyjnej. Jednocześnie uzyskano najdłuższy czas misji, wynoszący 822.8 s, oraz największy czas manewrów korekcyjnych, równy 31.32 s. Wariant ten eksplorował również najwięcej węzłów. Wyniki te wskazują, że FCN_ResNet50 zwiększa bezpieczeństwo, ale może prowadzić do bardziej konserwatywnego zachowania robota.

Najkorzystniejszy kompromis pomiędzy bezpieczeństwem a płynnością ruchu uzyskano dla wariantu $A^* + v$ Transformer. Podobnie jak wariant z FCN_ResNet50, metoda ta wyeliminowała kolizje, ale jednocześnie osiągnęła najmniejszą rzeczywistą długość ścieżki, najmniejsze średnie i maksymalne odchylenie od planu, najmniejszą liczbę skrętów oraz najmniejszą liczbę zatrzymań. Czas manewrów korekcyjnych był również znacznie krótszy niż w przypadku wariantu z FCN_ResNet50. Oznacza to, że model transformerowy pozwolił zachować bezpieczeństwo przejazdu bez tak dużych opóźnień w porównaniu z siecią konwolucyjną.

Na podstawie uzyskanych wyników można stwierdzić, że zastosowanie FCN_ResNet50 jako modułu percepcji wizyjnej poprawia bezpieczeństwo względem klasycznego algorytmu A^* , jednak odbywa się to kosztem dłuższego czasu przejazdu i większej liczby korekt. Model transformerowy daje bardziej zrównoważone działanie, ponieważ łączy brak kolizji z płynniejszym przebiegiem trajektorii.

5. Wnioski

W artykule przedstawiono porównanie działania dwóch modułów percepcji wizyjnej wspomagających planowanie trajektorii robota mobilnego: FCN_ResNet50 oraz modelu transformerowego SegFormer-B0. Oba rozwiązania zestawiono z klasycznym wariantem algorytmu A^* korzystającym z mapy zajętości środowiska.

Uzyskane wyniki wskazują, że zastosowanie percepcji wizyjnej może poprawić bezpieczeństwo nawigacji względem klasycznego algorytmu A^* . Wariant bazowy osiągnął najkrótszy czas misji, jednak przejazd zakończył się kolizjami. Warianty wykorzystujące segmentację obrazu pozwoliły wyeliminować kolizje, co potwierdza zasadność uzupełniania klasycznego planowania o informację pochodzącą z kamery.

Porównanie FCN_ResNet50 i modelu transformerowego pokazuje jednak istotne różnice w zachowaniu robota. Wariant z FCN_ResNet50 był bardziej zachowawczy, co przełożyło się na dłuższy czas realizacji misji oraz większy czas manewrów korekcyjnych. Może to wynikać z mniej stabilnej lub zbyt szerokiej interpretacji obszarów uznawanych za przeszkody.

Najkorzystniejszy kompromis pomiędzy bezpieczeństwem a płynnością ruchu uzyskano dla wariantu z modelem transformerowym. W badanym scenariuszu metoda ta pozwoliła uniknąć kolizji, ograniczyć liczbę skrętów i zatrzymań oraz zmniejszyć maksymalne odchylenie od planowanej ścieżki. Wyniki sugerują, że architektury transformerowe mogą być bardziej przydatne niż klasyczne sieci konwolucyjne w zadaniach wymagających stabilnej segmentacji otoczenia robota.

Przedstawione badania należy traktować jako analizę porównawczą przeprowadzoną w ograniczonym środowisku symulacyjnym. Dalsze prace powinny obejmować szerszą walidację eksperymentalną, większą liczbę przejazdów oraz testy w zróżnicowanych środowiskach.

LITERATURA

1. Dijkstra E. W.: A note on two problems in connexion with graphs, *Numerische Mathematik*, 1959, vol. 1 p. 269–271.
2. Dosovitskiy A., others : An image is worth 16x16 words: Transformers for image recognition at scale, In: *International Conference on Learning Representations*, 2021,.
3. Gu J., Kwon H., Wang D., Ye W., Li M., Chen Y.-H., Lai L., Chandra V., Pan D. Z.: Multi-scale high-resolution vision transformer for semantic segmentation, In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022,.
4. Hart P. E., Nilsson N. J., Raphael B.: A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, 1968, vol. 4(2) p. 100–107.
5. Li G.-H., Chen S.-L.: Visual slam in dynamic scenes based on object tracking and static points detection, *Journal of Intelligent and Robotic Systems*, 2022, vol. 104 p. 33.
6. Long J., Shelhamer E., Darrell T.: Fully convolutional networks for semantic segmentation, In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, p. 3431–3440.
7. Melo de M. P., Cambuim L. F. S., Barros E.: Occupancy grid map estimation based on visual slam and ground segmentation, In: *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*. IEEE, 2021, p. 288–293.
8. Ren J., others : Slam, path planning algorithm and application research of an indoor substation wheeled robot navigation system, *Electronics*, 2022, vol. 11(12) p. 1838.
9. Szwedka M., Budzan S.: Improving the performance of heuristic path planning algorithms by a computer vision obstacle detection method, In: *IEEE Proceedings of 2025 Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, DOI: 10.23919/SPA65537.2025.11215151 Poznan, Poland 2025, p. 50–55.
10. Tang Y., Zakaria M. A., Younas M.: Path planning trends for autonomous mobile robot navigation: A review, *Sensors*, 2025, vol. 25(4) p. 1206.
11. Xie E., Wang W., Yu Z., Anandkumar A., Alvarez J. M., Luo P.: Segformer: Simple and efficient design for semantic segmentation with transformers, *Advances in Neural Information Processing Systems*, 2021, vol. 34 p. 12077–12090.
12. Zhang L., Lu J., Zheng S., Zhao X., Zhu X., Fu Y., Xiang T., Feng J., Torr P. H. S.: Vision transformers: From semantic segmentation to dense prediction, *International Journal of Computer Vision*, 2024, vol. 132 p. 6142–6162.