

Automatyczna diagnostyka elementów toru kolejowego z użyciem sieci neuronowych głębokiego uczenia

Piotr Bojarczak, Waldemar Nowakowski

Uniwersytet Radomski im. Kazimierza Pułaskiego, ul. Malczewskiego 29, 26-600 Radom

Streszczenie: W artykule omówiono metodę automatycznej diagnostyki szlaku kolejowego. Polega ona na automatycznej ocenie stanu technicznego wybranych elementów toru, takich jak szyny, podkłady drewniane, podkłady betonowe, przytwierdzenia podkładów oraz rozjazdy. Przeprowadzono ją na podstawie analizy obrazów wideo elementów toru kolejowego zarejestrowanych przez dwie kamery liniowe umieszczone na dreźnie pomiarowej. Do oceny stanu technicznego badanych elementów zastosowano wybraną sieć neuronową głębokiego uczenia FCN-8. Określono też skuteczność zastosowanego algorytmu na podstawie takich miar jak IoU, Precision, Recall. Przedstawiono wnioski dotyczące zastosowania sieci FCN-8 w automatycznej klasyfikacji cech wybranych elementów toru kolejowego. Uzyskane rezultaty porównano z innymi metodami wykorzystywanymi w diagnostyce wizyjnej.

Słowa kluczowe: diagnostyka wizyjna toru kolejowego, algorytmy uczące, przetwarzanie obrazów

1. Wprowadzenie

Stan toru kolejowego ma istotny wpływ na bezpieczeństwo jazdy pociągów. Jest to szczególnie istotne przy zwiększających się prędkościach pociągów oraz stałym wzroście masy towarów i liczby pasażerów. Wiele wypadków kolejowych jest spowodowanych złym stanem elementów torów, takich jak podkłady i elementy mocujące szynę do podkładów kolejowych. Każdy wypadek generuje straty finansowe. Dlatego ciągle monitorowanie infrastruktury torowej jest niezmiernie ważne dla zapewnienia bezpieczeństwa ruchu [27]. Kontrolę stanu szyn sieci PKP realizuje się głównie metodami ultradźwiękowymi [23]. Stosuje się w tym celu wagon z zamontowaną zautomatyzowaną aparaturą do defektoskopii ultradźwiękowej. Inne elementy toru kolejowego, takie jak podkłady i przytwierdzenia szyn kontroluje się obecnie w Polsce bez użycia specjalistycznej aparatury pomiarowej. W uzasadnionych przypadkach przeprowadza się ręczne badania wybranych odcinków toru. Wykonuje je przeszkolony w tym celu personel, wyposażony w ręczne defektoskopy ultradźwiękowe. Jest to jednak sposób mało efektywny, ponieważ diagnosta może skontrolować tylko niewielki odcinek

toru. Ponadto skuteczność takiego sprawdzania zależy też od indywidualnych predyspozycji pracownika.

Badania torów kolejowych obejmują następujące czynności [11]:

- wykrywanie wad powierzchniowych w główce szyny,
- wykrywanie uszkodzeń przytwierdzeń szyn do podkładów,
- wykrywanie pęknięć w podkładach,
- pomiar szerokości toru,
- diagnostyka stanu podsypki.

Rozwój metod wizyjnych oraz algorytmów sztucznej inteligencji spowodował, że do powyższych zadań coraz częściej stosuje się nowe metody i technologie. Ogromny postęp w rozwoju przetworników wizyjnych CMOS i procesorów graficznych zaowocował powstaniem komercyjnych kamer liniowych i wysoce wydajnych serwerów graficznych. Można je stosować jako elementy systemu wizyjnego do kontroli elementów torów kolejowych. System taki można zainstalować m.in. na wagonie ultradźwiękowym lub wagonie do pomiaru geometrii torów. Stanowi on bazę do implementacji zaawansowanych algorytmów wizualnych.

W artykule przedstawiono metodę pozwalającą na zautomatyzowaną detekcję stanu technicznego elementów toru kolejowego. Do jej realizacji autorzy zastosowali w pełni konwulucyjną sieć z 8-krotnym upsamplingiem (FCN-8) do segmentacji obrazu torów kolejowych na obszary zawierające podsypkę, podkłady, łączniki, szyny i rozjazdy. System może rozróżnić podkłady drewniane i betonowe. Może również wykrywać cztery rodzaje przytwierdzeń szyn do podkładów, tj.: SB, KS, W-14, K (rys. 1). Skuteczność proponowanego rozwiązania przetestowano też dla różnych poziomów jasności obrazów toru kolejowego.

Autor korespondujący:

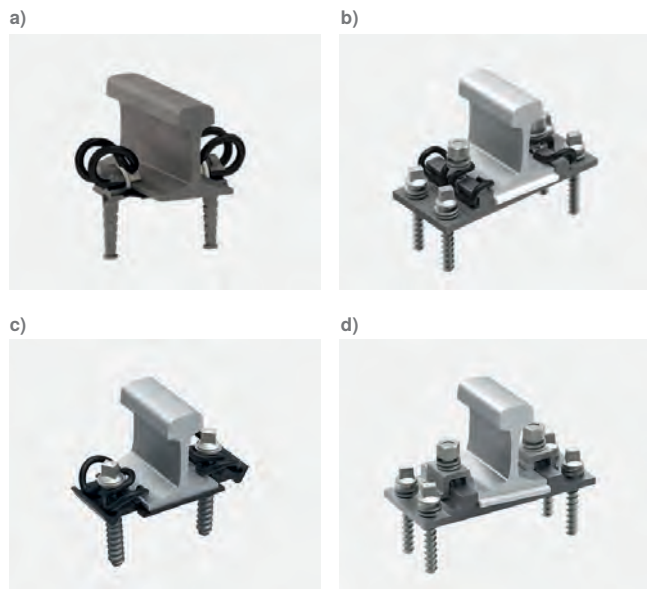
Piotr Bojarczak, p.bojarczak@uthrad.pl

Artykuł recenzowany

nadesłany 20.01.2024 r., przyjęty do druku 22.07.2024 r.



Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0



Rys. 1. Typy przytwierdzeń szyn do podkładów kolejowych: a) typ SB (Przytwierdzenie 1), b) typ KS (Przytwierdzenie 2), c) typ W-14 (Przytwierdzenie 3), d) typ K (Przytwierdzenie 4) [39]
 Fig. 1. Types of rail fastenings to railroad sleepers: a) type SB (Fastening 1), b) type KS (Fastening 2), c) type W-14 (Fastening 3), d) type K (Fastening 4) [39]

2. Aktualny stan badań

W ciągu ostatnich dwóch dekad systemy wizyjne stały się łatwo dostępne i znalazły zastosowanie również w kolejnictwie. Systemy te początkowo jedynie rejestrowały obrazy szlaku kolejowego przeznaczone do sprawdzenia przez przeszkolony personel. Następnie pojawiły się bardziej zaawansowane systemy. Istnieje wiele artykułów opisujących algorytmy do wykrywania elementów toru kolejowego. W pracy [1] przedstawiono algorytm do wykrywania przytwierdzeń podkładów do szyn. Wykorzystuje on banki filtrów. Analizowany obraz jest filtrowany indywidualnie przez każdy filtr. Rodzaj przytwierdzenia do podkładów wykrywa się przyporządkowanym mu filtrem. W tym celu zastosowano filtr korelacyjny (OT-MACH) (ang. *Optimal Trade-off Maximum Average Correlation Height*) [22]. Występowanie przytwierdzenia w analizowanym obrazie poszukuje się przez porównywanie fragmentu obrazu ze wzorcem przytwierdzenia jako uśrednioną postacią spektrum elementu jego obrazu. Uśrednianie przeprowadza się dla wielu obrazów przytwierdzenia. Na bieżąco oblicza się statystykę określającą średnią oraz wariancję odległości między znajdującymi się obok siebie wykrytymi przytwierdzeniami. Gdy odległość między ostatnio wykrytym przytwierdzeniem (wykrywanie przy pomocy filtru OT-MACH) a jego sąsiadem odbiega od tej statystyki, wówczas oznacza to, że między nimi znajduje się przytwierdzenie niewykryte przez algorytm. Zaproponowany algorytm pozwolił na wykrywanie przytwierdzeń z dokładnością ponad 90 %. Metody, które rozpoznają takie elementy toru jak podkłady kolejowe, elementy przytwierdzeń – podkładki, kolce i kotwy, autorzy przedstawili w pracy [28].

Podkłady i podkładki wykrywa się przy użyciu filtru krańcowego wraz z transformatą Hougha. Kolce i kotwy rozpoznaje się za pomocą algorytmu AdaBoost. W pracy [14] wykorzystuje się falki Haara jako cechy opisujące elementy złączne na obrazie toru. Wykrywa się je za pomocą klasyfikatora (detektora) zbudowanego przy użyciu probabilistycznego modelu graficznego (PGM). W pracy [31] zaproponowano algorytm wykrywający uszkodzenia przytwierdzeń pośrednich. W tym celu stosuje się transformację falkową

oraz PCA (ang. *Principal Component Analysis*) do ekstrakcji cech. Do detekcji (klasyfikacji) przytwierdzeń użyto sieci RBF (ang. *Radial Basis Function*) oraz MLP (ang. *Multilayer Perceptron*).

Inne podejście do wykrywania przytwierdzeń typu VOSSLOH przedstawia praca [37]. Natomiast w pracy [17] wykorzystano jako cechy histogram zorientowanych gradientów (HOG), a maszynę wektorów nośnych SVM (ang. *Support Vector Machine*) jako klasyfikator do wykrywania elementów złącznych na obrazie. Algorytm zaprezentowany w [19] umożliwia wykrywanie szyn, podkładów i płyt przytwierdzeń. Wykorzystano w tym celu filtr Canny'ego, transformatę Hougha i operacje morfologiczne do ekstrakcji cech oraz drzewo decyzyjne (algorytm C4.5) jako klasyfikator. W pracy [3], do zlokalizowania szyny na obrazie wykorzystuje się rozkład średniej energii współczynników szczegółów pionowych dla czwartego poziomu dekompozycji transformaty falkowej 2D obrazu. W pracy [35] wykorzystano bank filtrów Gabora do ekstrakcji cech i SVM jako klasyfikator do segmentacji podkładów drewnianych od podsypki. W powyższych artykułach cechy są wyodrębniane i wybierane ręcznie. Ponadto detektor zawiera dwa oddzielne etapy działania: ekstrakcji cech i budowy klasyfikatora. Proces taki nazywany jest uczeniem płytkim. Głównym problemem jest tu ograniczony zbiór cech, z którego są one wyodrębniane. Z kolei może to czasami prowadzić do selekcji cech, które nieprecyzyjnie opisują wykrywane obiekty. Przełomem w dziedzinie uczenia maszynowego było pojawienie się pracy dotyczącej zastosowania głębokiej sieci konwolucyjnej w detekcji obiektów [21]. Głęboka sieć neuronowa tworzy zbiór cech o rozmiarze znacznie większym niż dla płytkiego uczenia, praktycznie ograniczonego rozmiarem analizowanego obrazu. Ponadto ekstrakcja cech i budowa klasyfikatora są wykonywane w jednym etapie podczas procesu uczenia, a cechy są wybierane niezależnie od użytkownika. Od tego czasu wielu badaczy próbowało zastosować głębokie sieci neuronowe do wizyjnej inspekcji torów kolejowych. Na przykład w [16] stosuje konwolucyjną sieć składającą się z czterech warstw konwolucyjnych do monochromatycznych obrazów torów kolejowych. Umożliwia ona wykrywanie 10 rodzajów tekstur, w tym podsypki, szyn, podkładów drewnianych i betonowych. W pracy [15] do segmentacji obrazu wykorzystuje się również sieć konwolucyjną składającą się z czterech warstw. Dodatkowo rozpoznaje przytwierdzenia szyn i klasyfikuje je na klasy zawierające egzemplarze normalne i uszkodzone przytwierdzenia. Klasyfikację przeprowadza się przez połączenie sieci konwolucyjnej z SVM. W pracy [32] użyto wstępnie wytrenowanej szybszej sieci R-CNN do przeprowadzenia segmentacji analizowanego obrazu na powierzchnie toru i na przytwierdzenia. Segmentację tę wykorzystuje się do identyfikacji anomalii. Następnie zastosowano GAN (ang. *Generative Adversarial Network*) do grupowania normalnych i anormalnych obserwacji. Praca [12] przedstawia zastosowanie sieci głębokich (AlexNet i ResNet) w rozpoznawaniu defektów przytwierdzeń.

W ostatnich kilku latach opublikowano wiele artykułów dotyczących zastosowania głębokich sieci neuronowych w wykrywaniu obiektów na obrazach. Na przykład w [33] przedstawiono sieć You Only Look Once (YOLO). Wykorzystuje ona do detekcji obiektów w obrazach zarówno informacje kontekstowe, jak i związane z ich wyglądem. Pozwala to na przetwarzanie obrazów z szybkością do 50 klatek na sekundę. Inną szybką sieć SSD (ang. *Single Shot Multi-Box Detector*) przedstawiono w pracy [29]. Pozwala ona na detekcję obiektów z prędkością do 58 klatek na sekundę. Kolejna sieć [34] o nazwie Faster R-CNN składa się z dwóch podsieci: Region Proposal Network (RPN) i drugiej sieci, która wykonuje ostateczną detekcję obiektu w obszarze

wskazany przez RPN. Pomimo wysokiej precyzji wykrywania, przetwarza ona obrazy ze stosunkowo małą szybkością 5 kl./s. We wszystkich powyższych sieciach pozycję wykrytego obiektu oznacza się ograniczającą ramką wyśrodkowaną na tym obiekcie. Sieć Mask R-CNN [18], oprócz generowania obwiedni, tworzy również maskę dla wykrytego obiektu. Cały proces wykrywania jest podzielony na dwa etapy. Najpierw sieć o architekturze bardzo podobnej do Faster R-CNN wykrywa obiekt i zaznacza go obwiednią. Następnie inna sieć przeprowadza segmentację semantyczną w obszarze ograniczonym do tej obwiedni. W rezultacie każdemu punktowi w obwiedni przypisywana jest etykieta obiektu, jeśli do niego należy, lub etykieta tła, jeśli do niego nie należy. Sieć Mask R-CNN działa z szybkością do kilku klatek na sekundę. Inną grupę stanowią sieci stosowane wyłącznie do semantycznej segmentacji obrazów. Obejmuje ona sieci o nazwach: DeepLabv1 [8], DeepLabv2 [7], DeepLabv3 [9], DeepLabv3Plus [10] oraz Fully Convolutional Network (FCN) [30]. Architektury sieci DeepLabv1 i DeepLabv2 są podobne. Obie wykorzystują sieć Deep Convolutional Network wraz z Atrous Convolution i Fully Connected Conditional Random Field (CRF) do segmentacji obrazu. Dodatkowo sieć DeepLabv2 jest wyposażona w moduł Atrous Spatial Pyramid Pooling (ASPP), który pozwala zwiększyć dokładność segmentacji. ASPP integruje cechy z wielu skal. Dzięki temu można łatwo zidentyfikować obiekty o różnych wymiarach. DeepLabv3 – to zmodyfikowana wersja DeepLabv2, w której Atrous zastąpiono pojedynczym modułem ASPP. W DeepLabv3 usunięto również moduł CRF. Aby bardziej zwiększyć dokładność, w najnowszej wersji o nazwie DeepLabv3plus dodano moduł dekodera. Mimo stosunkowo wysokiej precyzji wykrywania, przetwarzają one obraz z szybkością 8 FPS, co sprawia, że nie mogą działać w czasie rzeczywistym.

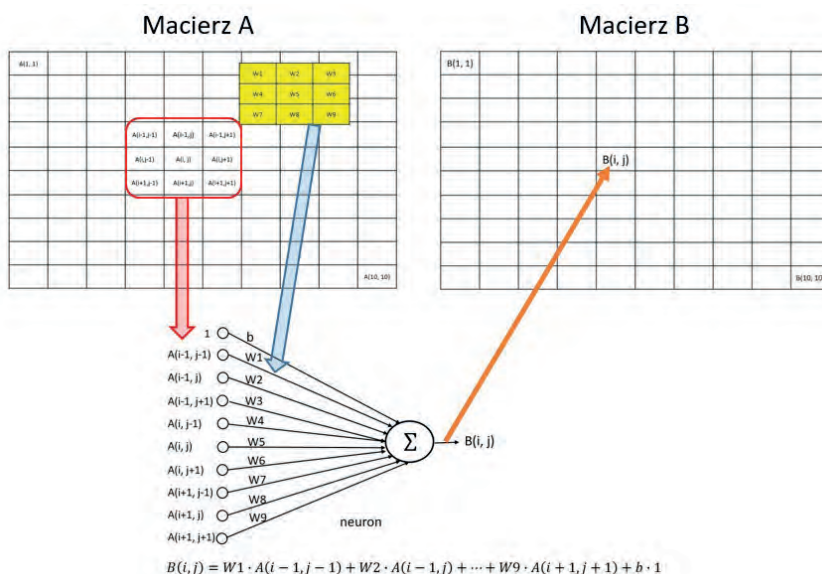
Nabyte przez autorów doświadczenia w przetwarzaniu obrazów, w budowie i działaniu sieci neuronowych oraz w specyfice zagadnień diagnostyki toru kolejowego, zaowocowały powstaniem szeregu prac o wykorzystaniu, przetwarzaniu oraz analizie sygnałów i obrazów w diagnostyce toru kolejowego. W pracy [24] przedstawiono zastosowanie sieci SVM wraz z transformacją falkową oraz przekształcenie

PCA (ang. *Principal Component Analysis*) do wykrywania wad typu *head checking* na podstawie sygnałów uzyskanych w oparciu o metodę magnetycznej pamięci metalu. W pracy [4] zaprezentowano wykorzystanie filtrów entropijnych oraz sieci neuronowych do określania stanu drewnianych podkładów kolejowych. Kolejne prace [5, 6] przedstawiają sposób detekcji przytwierdzeń szyn do podkładów na podstawie ich obrazów z wykorzystaniem sieci SVM oraz transformacji falkowej. W [2] przedstawiono metodę ekstrakcji szyny z obrazu, która stanowi element składowy algorytmów wizyjnej diagnostyki toru kolejowego. Natomiast w [25] zaprezentowano wykorzystanie skatometrii laserowej wraz z transformacją falkową oraz teorią zbiorów rozmytych do detekcji wad typu *head checking*. W pracy [27] zaprezentowano inteligentne klasyfikatory do klasyfikacji wad kontaktowo-naprężeniowych w szynach kolejowych. W kolejnej pracy [26] przedstawiono algorytmy przetwarzania oraz klasyfikacji obrazów ultradźwiękowych pochodzących z badań metodą TOFD.

3. Metoda badawcza

W proponowanym rozwiązaniu struktury systemu diagnostycznego możliwe jest wykrywanie elementów toru i określanie stopnia pokrycia podkładu przez podsypkę. Użyto w nim sieci konwolucyjnej z 8-krotnym powiększeniem (dalej nazywanym w artykule upsamplingiem) do oryginalnego rozmiaru obrazu, która nazywa się FCN-8 [30]. Przeprowadza ona semantyczną segmentację obrazu. Podczas tego procesu sieć przypisuje etykietę do każdego piksela w obrazie. Jej wartość zależy od obiektu, do którego należy piksel. Główną strukturą tej sieci jest konwolucyjna sieć neuronowa (CNN). Istnieją trzy główne architektury CNN: AlexNet, VGGNet i GoogLeNet. Według [30] VGGnet ma najwyższą dokładność segmentacji spośród wymienionych tu sieci. Dlatego też została ona wybrana do realizacji naszego systemu. Sieć VGGNet, zwana też VGG-16, składa się ze stosu warstw konwolucyjnych, ReLu i Pooling, po których następują trzy w pełni połączone warstwy. Warstwa konwolucyjna wykonuje operację splotu (konwolucji) zdefiniowaną jako:

$$B(i, j, z) = \left[\sum_{k=-\frac{K-1}{2}}^{\frac{K-1}{2}} \sum_{m=-\frac{K-1}{2}}^{\frac{K-1}{2}} \sum_{n=0}^N A(i+k, j+m, n) \cdot W\left(k + \frac{K-1}{2}, m + \frac{K-1}{2}, n, z\right) \right] + b(z) \quad (1)$$



gdzie: $i \in (0, \dots, P-1)$, $j \in (0, \dots, R-1)$, $z \in (0, \dots, Z-1)$ – współrzędne macierzy wyjściowej B warstwy odpowiednio wzdłuż osi X, Y, Z ; A – macierz wejściowa warstwy o wymiarze $P \times R \times N$; W – macierz wag o wymiarach $K \times K \times N \times Z$.

Rys. 2. Przykład splotu (konwolucji) dla macierzy A o wymiarze $10 \times 10 \times 1$ oraz macierzy wag W o wymiarze $3 \times 3 \times 1 \times 1$
Fig. 2. Example of convolution for matrix A of size $10 \times 10 \times 1$ and weight matrix W of size $3 \times 3 \times 1 \times 1$

P1	K1	K2	P2
K3	K4	K5	K6
K7	K8	K9	K10
P3	K11	K12	P4

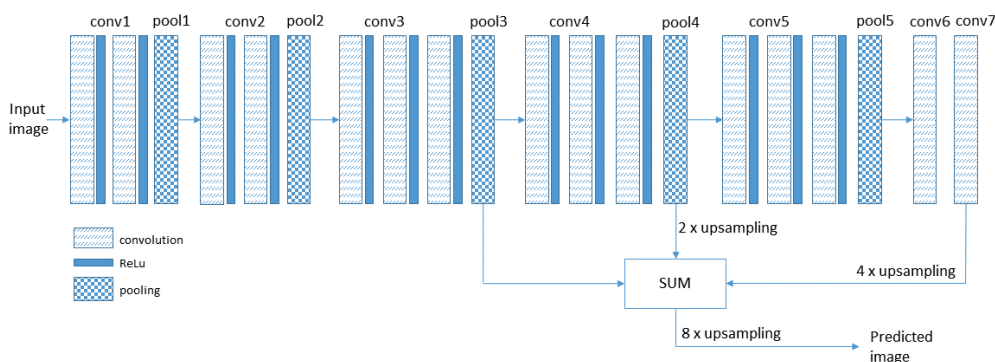
P1, P2, P3, P4 - intensywności najbliższych punktów obrazu oryginalnego
 K1, K2, ..., K11, K12 - intensywności dodatkowych punktów uzyskanych w oparciu o interpolację bilinearną

Rys. 3. Idea operacji 4-krotnego upsamplingu
 Fig. 3. Illustration of 4 times upsampling

K określa szerokość filtra/jądra, Z odpowiada liczbie filtrów używanych w tej warstwie, b jest wektorem odchylenia (ang. *bias*) o wymiarze Z . Podmacierz macierzy W o wymiarze $K \times K \times N$ odpowiada pojedynczemu neuronowi, dołączonemu do ograniczonej liczby wejść określonych przez $K \times K \times N$. Warstwa ta wykonuje operację $B = f(A, W)$ przekształcającą macierz A o wymiarze $P \times R \times N$ w macierz B o wymiarze $P \times R \times Z$. W przypadku pierwszej warstwy sieci macierz A odpowiada obrazowi wejściowemu i $N = 3$ (składowe R, G, B obrazu).

Macierz wyjściowa B poddawana jest nieliniowej funkcji aktywacji *ReLU* w postaci $C = \max(0, B)$. Dodatkowo, co kilka warstw konwolucyjnych macierzy C o wymiarze $P \times R \times Z$ redukuje się do $P/2 \times R/2 \times Z$ przez podpróbkowanie lub łączenie [30]. Ostatnie trzy warstwy, to warstwy w pełni połączone. Oznacza to, że każdy neuron tej warstwy jest połączony ze wszystkimi wejściami poprzedniej warstwy. Ostatnia warstwa składa się z 1000 neuronów. Każdy neuron w tej warstwie jest odpowiedzialny za wskazanie jednej klasy obiektów występujących w analizowanym obrazie. Według [30], w pełni połączoną warstwę można postrzegać jako warstwę konwolucyjną z filtrem/jądrem, która pokrywa całe regiony wejściowe. W rezultacie tworzy się zgrubną mapę o rozmiarze $H/32 \times G/32 \times 1000$, gdzie H i G oznaczają odpowiednio wysokość i szerokość analizowanego obrazu. Najprostsza metoda polega na 32-krotnym upsamplingu mapy zgrubnej do rozmiaru oryginalnego obrazu. Jednak uzyskany wynik jest również bardzo zgrubny. Upsampling polega na zwiększeniu wymiaru oryginalnego obrazu przez umieszczenie między kolejnymi jego punktami dodatkowych punktów. Intensywność tych punktów jest określana na podstawie punktów oryginalnego obrazu z wykorzystaniem interpolacji bilinearnej.

Aby poprawić dokładność segmentacji, informacje pochodzące z kilku wcześniejszych warstw dodaje się do zgrubnej formy. Zgodnie z [13] operację upsamplingu można wykonać przy użyciu transponowanej konwolucji. FCN-8 sumuje $2 \times$ upsamplowaną warstwę conv7 z warstwą pool4. Następnie wynik jest upsamplowany dwa razy i sumowany z warstwą pool3. Ostateczna mapa segmentacji jest tworzona przez 8-krotny upsampling uzyskanego wyniku – rys. 4.



Rys. 4 Schemat blokowy sieci neuronowej FCN-8
 Fig. 4. Block diagram of FCN-8

Wartości wszystkich parametrów (wag) sieci określa się w trakcie procesu uczenia przez minimalizację następującej funkcji celu:

$$L = -\frac{1}{S} \sum_{i=1}^S P(x_i) \log Q(x_i) \quad (2)$$

gdzie:

$P(x_i)$ – prawdopodobieństwo określające, do której klasy należy piksel x_i . Ma ono postać wektora $P = [p_1, \dots, p_p, \dots, p_K]$. K jest liczbą klas (grup), na jakie klasyfikowany jest piksel x_i (tutaj $K = 9$: cztery rodzaje przytwardzeń + dwa rodzaje podkładów + szyna + rozjazd + tło). Do każdego elementu wektora przypisana jest z góry ustalona klasa. Na przykład, jeśli p_1 odpowiada klasie podkład betonowy wówczas, gdy piksel x_i należy do niej, element p_1 przyjmuje wartość 1, natomiast pozostałe elementy wartość 0. Wartości te są przypisywane w oparciu o dostępne dane uczące.

$Q(x_i)$ – prawdopodobieństwo określające, do której klasy należy piksel x_i . Jego postać jest taka sama jak w przypadku prawdopodobieństwa $P(x_i)$ – wektor $Q = [q_1, \dots, q_p, \dots, q_K]$ z tą różnicą, że elementy wektora nie są przypisywane na podstawie danych uczących (wartości 1 lub 0). Elementy tego wektora są generowane przez sieć dla aktualnych wartości jej wag. Aby zapewnić, że suma elementów generowanego przez sieć neuronową wektora Q jest równa 1 (suma wszystkich prawdopodobieństw wynosi 1), każde pojedyncze wyjście sieci neuronowej podlega normalizacji z wykorzystaniem funkcji *softmax* zdefiniowanej jako:

$$q_t = \frac{e^{y_t}}{\sum_{r=1}^K e^{y_r}} \quad (3)$$

gdzie q_t jest t -tym elementem wektora Q , zaś y_t jest t -tym wyjściem sieci neuronowej podlegającym normalizacji przez funkcję *softmax*.

S – całkowita liczba danych uczących przekazanych sieci podczas procesu uczenia.

Gdy funkcja celu jest bliska 0, oba rozkłady prawdopodobieństwa są prawie takie same, tj. $p(x) \approx q(x)$ i proces uczenia zostaje zatrzymany.

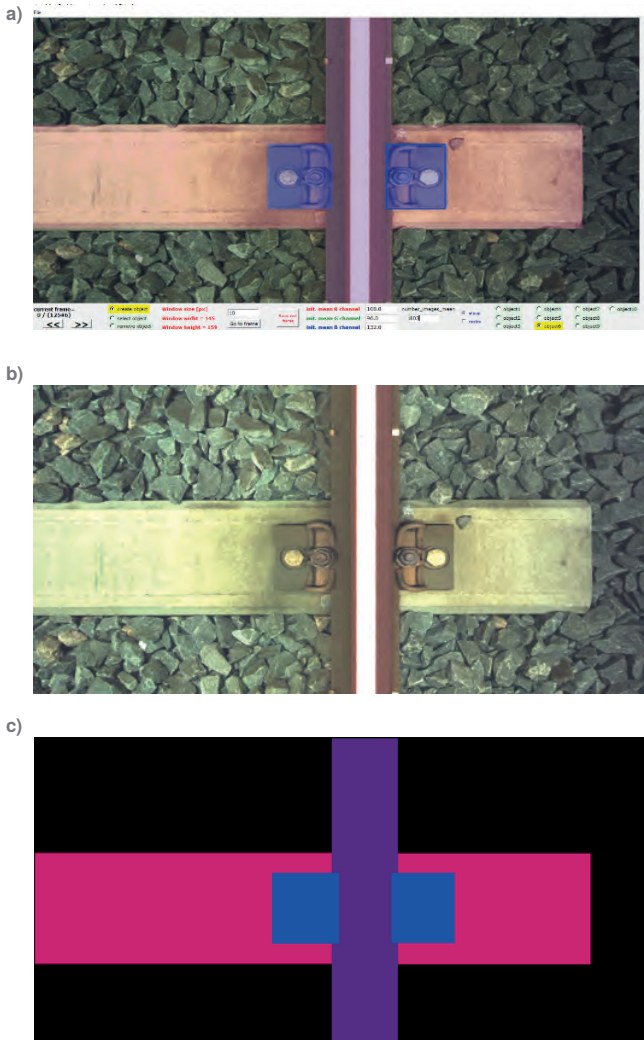
4. Wyniki eksperymentów

Na potrzeby badań zdecydowano się przyuczyć sieć już wytrenowaną do wykrywania elementów toru kolejowego. Proces ten, nazywany po angielsku *transfer learning*, oznacza, że model opracowany dla jednego zadania, jest ponownie wykorzystywany jako punkt wyjścia dla modelu w drugim zadaniu [36].

W taki sposób można znacznie przyspieszyć uczenie i poprawić skuteczność detekcji realizowanej przez zaproponowany model sieci. W tym celu użyto wstępnie wytrenowaną sieć VGG-16 jako bazę do opracowania systemu wizyjnego opartego na FCN-8. Sieć została zbudowana z użyciem pakietu Tensorflow.

4.1. Dane źródłowe

Obrazy torów kolejowych zostały zarejestrowane w postaci plików wideo. W tym celu użyto dwóch kamer liniowych, zamontowanych pod podłogą wagonu diagnostycznego. Kamery umieszczone są prostopadle do toru. Jedna kamera obejmuje lewą połowę toru, a druga prawą połowę toru.



Rys. 5. Program do anotacji obrazów, a) zrzut ekranu, b) obraz poddawany anotacji, c) obraz z zaznaczoną maską odpowiadającą obrazowi na rys. 5b)

Fig. 5. Image annotation program, a) screenshot, b) image being annotated, c) image with marked mask corresponding to the image in Fig. 5b)

4.2. Analiza wydajności sieci

Uczenie sieci przeprowadzono dla 400 epok¹ wykorzystując do tego celu kartę graficzną NVIDIA GeForce GTX 1080 Ti GPU. Do optymalizacji użyto pakietu AdamOptimizer, dla którego ustalono: współczynnik uczenia $Lr = 0,0001$, współczynnik wygaszania wag $w_dec = 0,0005$ oraz $momentum = 0,9$. Podczas uczenia wykorzystano dwie grupy danych: $train_data$ do szkolenia i $valid_data$ do testowania. Po uczeniu wydajność sieci zweryfikowano na zestawie danych testowych. Wykorzy-

stano tu cztery metryki, a mianowicie: dokładność pikseli (ang. *pixel accuracy*), przecięcie IoU (ang. *Intersection over Union*), precyzję (ang. *precision*) oraz przypomnienie (ang. *recall*). Dokładność pikseli oblicza się jako procent pikseli w obrazie, które poprawnie sklasyfikowano. Określa się ją za pomocą następującej zależności:

$$Pixel\ accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

gdzie:

TP (ang. *true positive*) – wynik prawdziwie pozytywny, reprezentuje piksele, co do których przewidziano, że należą do danej klasy (zgodnie z maską docelową (prawdy)),

TN (ang. *true negative*) – wynik prawdziwie negatywny – reprezentuje piksele, które zostały poprawnie zidentyfikowane jako nienależące do danej klasy,

FP (ang. *false positive*) – wynik fałszywie pozytywny – reprezentuje piksele, co do których przewidziano, że należą do danej klasy, choć do niej nie należą,

FN (ang. *false negative*) – wynik fałszywie negatywny – oznacza piksele, które zostały zidentyfikowane jako nienależące do danej klasy, choć do niej należą.

Druga metryka zwana Intersection over Union (IoU), określana również jako indeks Jaccarda, jest zasadniczo metodą ilościowego określania procentowego nakładania się maski docelowej (prawdy) i wyników predykcji. IoU określa liczbę pikseli wspólnych dla maski docelowej i predykcyjnej podzieloną przez całkowitą liczbę pikseli obecnych w obu maskach:

$$IoU = \frac{T_M \cap P_P}{T_M \cup P_P} = \frac{TP}{TP + FP + FN} \quad (5)$$

gdzie:

T_M – piksele należące do maski docelowej,

P_P – reprezentuje piksele przewidywane jako należące do maski docelowej.

Precyzja jest określona jako stosunek prawidłowo sklasyfikowanych pikseli należących do danej klasy, do całkowitej liczby pikseli sklasyfikowanych jako należące do tej klasy:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Przypomnienie jest określone jako stosunek prawidłowo sklasyfikowanych pikseli należących do danej klasy do całkowitej liczby pikseli należących do niej:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

W tabeli 1 przedstawiono średnie wartości tych czterech metryk obliczone dla elementów toru kolejowego. Wartości te zostały określone na podstawie zestawu $test_data$ danych testowych. Średni czas przetwarzania dla tego systemu wyniósł 67 ms.

Przeprowadzono też porównanie uzyskanych wyników z innymi pracami. W pracy [19] wykorzystano filtr Cannyego wraz z drzewami decyzyjnymi do detekcji elementów toru. Średnia wartość miary dokładności (ang. *pixel accuracy*) dla tego systemu wynosiła 0,83 (u nas: 0,891). Natomiast czas przetwarzania wyniósł 166 ms. W pracy [20] przedstawiono system do detekcji zwrotnic, w którym wykorzystano filtr Cannyego oraz

¹ Jedna epoka odpowiada pojedynczej aktualizacji wag sieci na podstawie aktualnej wartości funkcji błędów zdefiniowanej przez wzór (2)

Tabela 1. Wartości miar dla elementów toru kolejowego

Table 1. Measure values for railroad track elements

Element	Pixel Accuracy	IoU	Precision	Recall
Szyna	0,96	0,898	0,91	0,98
Podkład drewniany	0,876	0,861	0,89	0,96
Podkład betonowy	0,965	0,904	0,905	0,99
Przytwierdzenie 1	0,974	0,664	0,84	0,76
Przytwierdzenie 2	0,845	0,738	0,87	0,83
Przytwierdzenie 3	0,855	0,818	0,86	0,94
Przytwierdzenie 4	0,765	0,643	0,77	0,79
Rozjazd	0,923	0,903	0,91	0,99

transformatę Hougha. Średnia wartość miary dokładności dla tego systemu wynosiła 0,88, natomiast średni czas przetwarzania wyniósł 221 ms.

Na rysunku 6 przedstawiono przykłady segmentacji obrazu dla różnych elementów toru. Pierwsza kolumna przedstawia obraz toru, druga obraz z nałożoną przewidywaną maską, a ostatnia – tę maskę. Każdej klasie wykrywanych pikseli przyporządkowany jest z góry ustalony kolor. Na przykład wszystkie piksele należące do klasy rozjazd są zawsze zaznaczane przez model kolorem brązowym, piksele należące do klasy podkład betonowy kolorem amarantowy zaś piksele należące do klasy podkład drewniany kolorem fioletowym. Proponowana sieć FCN-8 może segmentować obraz nawet wtedy, gdy tor jest częściowo pokryty śniegiem – rys. 6 wiersz 1. Ponadto sieć może wykryć sytuację, gdy podkład jest częściowo pokryty tłucznem – rys. 6 wiersz 3, wiersz 4. Przedstawiony system przetwarza pojedynczy obraz ze średnią szybkością 15 klatek na sekundę.

Zaproponowany algorytm pozwala na wykrycie takich elementów infrastruktury kolejowej jak: szyna, cztery rodzaje przytwierdzeń, podkłady betonowe i drewniane oraz rozjazdy. Dodatkowo określenie stopnia pokrycia podkładów

przez tłuczeń pozwala na pośrednią ocenę stanu podtorza. Wykrycie stałych elementów infrastruktury takich jak rozjazdy umożliwia również cykliczną korekcję błędu położenia drewniny pomiarowej na szlaku. Zaproponowany system powinien potencjalnie umożliwić wykrywanie uszkodzeń przytwierdzeń szyn do podkładów oraz podkładów. Aby to zrealizować należy wprowadzić do danych uczących dodatkowe obrazy zawierające klasy reprezentujące różnego typu uszkodzenia. Liczba dostępnych przykładów uszkodzeń przytwierdzeń oraz podkładów jest ograniczona, co w przypadku modelu opartego na sieciach głębokich wymaga zgromadzenia ich większej ilości bądź też sztucznego zwiększenia ich wolumenu przez wykorzystanie metod opisanych w [38]. Będzie to kolejnym etapem naszych badań.

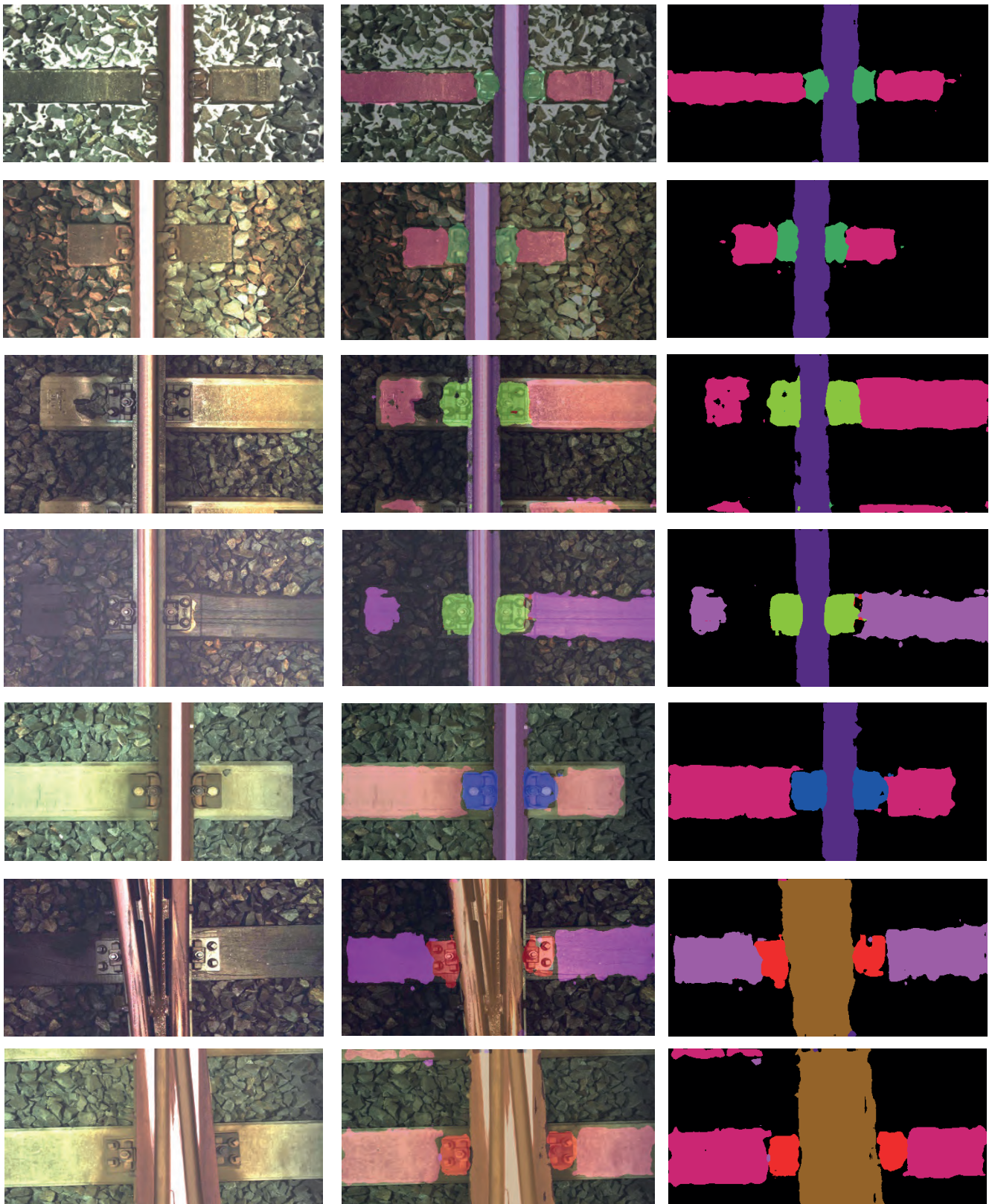
4.3. Analiza niezmienności jasności obrazu

Sprawdzono odporność algorytmu na zmiany jasności obrazu. W tym celu sztucznie zmieniano jasność każdego obrazu należącego do zbioru *test_data*. Osiągnięto to przez konwersję modelu obrazu RGB na model HSV (ang. *Hue, Saturation, Value*), w którym składowa *V* odpowiada za jasność obrazu. Jasność każdego obrazu zmieniono w następujący sposób:

Tabela 2. Wartości względnej dokładności dla różnych mnożników S

Table 2. Relative values of pixel accuracy for different multipliers S

Mnożnik S	S=0,1	S=0,2	S=0,3	S=0,4	S=0,5	S=0,6	S=0,7	S=0,8	S=0,9	S=1,0
Szyna	0,26	0,45	0,51	0,54	0,93	0,95	0,96	0,98	0,99	1,0
Podkład drewniany	0,34	0,73	0,76	0,81	0,89	0,92	0,97	0,98	0,98	1,0
Podkład betonowy	0,44	0,77	0,82	0,84	0,86	0,86	0,90	0,95	0,98	1,0
Przytwierdzenie 1	0,06	0,25	0,51	0,65	0,86	0,89	0,93	0,93	0,97	1,0
Przytwierdzenie 2	0,06	0,21	0,47	0,60	0,84	0,89	0,93	0,95	0,97	1,0
Przytwierdzenie 3	0,00	0,10	0,33	0,46	0,75	0,94	0,97	0,98	0,99	1,0
Przytwierdzenie 4	0,0	0,23	0,50	0,67	0,77	0,86	0,92	0,96	0,97	1,0
Rozjazd	0,31	0,5	0,6	0,63	0,97	0,98	0,98	0,99	0,99	1,0



Rys. 6. Przykłady wykrywania elementów toru: wiersz 1) i wiersz 2) – przytwierdzenie 1, podkład betonowy; wiersz 3) – przytwierdzenie 2 i podkład betonowy; wiersz 4) – przytwierdzenie 2 i podkład drewniany; wiersz 5) – przytwierdzenie 3 i podkład betonowy; wiersz 6) – przytwierdzenie 4, podkład drewniany i rozjazd; wiersz 7) – łącznik 4, podkład betonowy i rozjazd

Fig. 6. Examples of detecting track elements: line 1) and line 2) – fastener1, concrete sleeper; line3) – fastener 2 and concrete sleeper; line 4) – fastener 2 and wooden sleeper; line 5) – fastener 3 and concrete sleeper; line 6) – fastener 4, wooden sleeper and turnout; line 7) – fastener 4 concrete sleeper and turnout

$$I = S \cdot I_{nom} \quad (8)$$

gdzie: S – mnożnik z zakresu $[0,1, 0,2, \dots, 0,9, 1,0]$, I_{nom} – jasność każdego piksela na oryginalnym (niezmodyfikowanym) obrazie.

Każdy zmodyfikowany zestaw *test_data* zawierający obrazy z predefiniowanym poziomem jasności przekazano wytrenowanej

sieci. Następnie obliczono średnią dokładność pikseli i wskaźniki IoU dla obrazów wygenerowanych przez sieć i powiązano je ze średnią dokładnością pikseli i wskaźnikami IoU uzyskanymi dla poziomu jasności I_{nom} ($S = 1$).

W tabeli 2 przedstawiono związek między względną dokładnością pikseli i mnożnikiem S . Natomiast w tabeli 3 – związek między względną metryką IoU i mnożnikiem S .

Tabela 3. Wartości względne IoU dla różnych mnożników S

Table 3. Relative values of IoUs for different multipliers S

Mnożnik S	S=0,1	S=0,2	S=0,3	S=0,4	S=0,5	S=0,6	S=0,7	S=0,8	S=0,9	S=1,0
Szyna	0,19	0,33	0,42	0,50	0,88	0,89	0,92	0,95	0,95	1,0
Podkład drewniany	0,34	0,60	0,73	0,79	0,86	0,9	0,95	0,96	0,97	1,0
Podkład betonowy	0,43	0,7	0,77	0,84	0,85	0,9	0,94	0,96	0,98	1,0
Przytwierdzenie 1	0,07	0,29	0,45	0,68	0,84	0,89	0,93	0,96	0,98	1,0
Przytwierdzenie 2	0,06	0,18	0,39	0,60	0,85	0,88	0,91	0,94	0,97	1,0
Przytwierdzenie 3	0,01	0,11	0,29	0,4	0,72	0,95	0,96	0,98	0,98	1,0
Przytwierdzenie 4	0,03	0,25	0,52	0,71	0,78	0,87	0,92	0,95	0,96	1,0
Rozjazd	0,36	0,41	0,62	0,63	0,90	0,95	0,96	0,98	0,98	1,0

5. Wnioski

Utrzymanie torów jest kluczowym zadaniem, które ma ogromny wpływ na bezpieczeństwo ruchu kolejowego. W tej pracy autorzy opisali wyniki badań weryfikujących możliwość zastosowania sieci FCN-8 do detekcji stałych elementów szlaku kolejowego, tj.: przytwierdzeń drewnianych i betonowych, szyn i rozjazdów kolejowych.

Detekcję tę realizuje się przez segmentację obrazu. Każdemu pikselowi na obrazie przypisuje się etykietę obiektu, do którego on należy. Pozwala to też na określenie stopnia pokrycia podkładu przez podsypkę. Obraz toru rejestrują dwie kamery liniijkowe. Każda z kamer obejmuje połowę szerokości obszaru toru i jest podłączona do osobnej wytrenowanej sieci FCN-8. Obraz toru jest przetwarzany równoległe przez dwie sieci FCN-8, z których pierwsza odpowiada za lewą połowę obrazu, a druga za prawą połowę. Dzięki temu obraz pojedynczej ścieżki przetwarza się z szybkością 15 klatek na sekundę. W pracy [15] analizowano obraz toru przepuszczany przez w pełni konwolucyjną sieć neuronową. Wynikiem końcowym jest zgrubna mapa o wymiarze 1/16 obrazu wejściowego.

W zastosowanym tu podejściu, sieć FCN-8 łączy informacje szczegółowe z wcześniejszych warstw z informacjami zgrubnymi z mapy zgrubnej, a wynik jest próbkowany ośmiokrotnie. W rezultacie przewidywany obraz maski ma wymiar obrazu wejściowego zamiast 1/16 obrazu wejściowego [15].

Przetestowaliśmy również odporność naszego systemu na zmiany jasności obrazu. Test wykazał, że wykrywanie elementów przytwierdzeń jest umiarkowanie wrażliwe na zmianę jasności. Spadek poziomu jasności do 0,7 dla elementów złącznych powoduje spadek ich względnej dokładności pikseli do 0,9. Pozostałe elementy torów kolejowych są znacznie mniej wrażliwe na zmianę jasności. Spadek poziomu jasności do 0,3 powoduje spadek względnej dokładności IoU do 0,9.

Prezentowany tu system może wykrywać elementy torów kolejowych z szybkością 15 klatek na sekundę. Stanowi on użyteczną alternatywę szeregu innych istniejących systemów.

Bibliografia

1. Babenko P., *Visual inspection of railroad tracks*. PhD thesis, University of Central Florida, 2009.
2. Bojarczak P., *Application of wavelet transform into precise localization of railway rail edges in visual diagnostic of*

track, "Archives of Transport", Vol. 24, No. 1, 2012, 5–16, DOI: 10.2478/v10174-012-0001-9.

3. Bojarczak P., *Visual algorithms for automatic detection of squat flaws in railway rails*, "Insight – Non-Destructive Testing and Condition Monitoring", Vol. 55, No. 7, 2013, 353–359, DOI: 10.1784/insi.2012.55.7.353.
4. Bojarczak P., Lesiak P., *Application of neural networks into automatic visual diagnostic of railway wooden sleepers*, Międzynarodowa Konferencja Naukowa Transport XXI w., Białowieża 2010, "Logistyka", Nr 4, 2010.
5. Bojarczak P., Lesiak P., *SVM based classification method of railway's defects*, "Pomiary Automatyka Kontrola", R. 53, Nr 12, 2007, 15–17.
6. Bojarczak P., Lesiak P., *Visual system diagnosing the state of elements fastening the rail to the sleepers*. "Pomiary Automatyka Kontrola", R. 57, Nr 12, 2011, 1605–1607.
7. Chen L.C., Papandreou G., Kokkinos I., Murphy K., Yuille A.L., *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution and Fully Connected CRFs*, "Computer Vision and Pattern Recognition", 2018, DOI: 10.48550/arXiv.1606.00915.
8. Chen L.C., Papandreou G., Kokkinos I., Murphy K., Yuille A.L., *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs*, "Computer Vision and Pattern Recognition", 2014, DOI: 10.48550/arXiv.1412.7062.
9. Chen L.C., Papandreou G., Schroff F., Adam H., *Rethinking Atrous Convolution for Semantic Segmentation*, "Computer Vision and Pattern Recognition", 2017, DOI: 10.48550/arXiv.1706.05587.
10. Chen L.C., Papandreou G., Schroff F., Adam H., *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. "Computer Vision and Pattern Recognition", 2018, DOI: 10.48550/arXiv.1802.02611.
11. Camargo L., Edwards J., Barkan C., *Emerging condition monitoring technologies for railway track components and special track work*. ASME/ASCE/IEEE Joint Rail Conference, Pueblo, Colorado, 2011, DOI: 10.1115/JRC2011-56113.
12. Dai P., Du X., Wang S., Gu Z., Ma Y., *Rail fastener automatic method in complex background*. Proceedings of the SPIE, Vol. 10806, 2018, DOI: 10.1117/12.2503323.

13. Dumoulin V., Visin F., *A guide to convolution arithmetic for deep learning*, "Machine Learning", 2016, DOI: 10.48550/arXiv.1603.07285.
14. Feng H., Jiang Z., Xie F., Yang P., Shi J., Chen L., *Automatic fastener classification and defect detection in vision-based railway inspection systems*, "IEEE Transactions on Instrumentation and Measurement", Vol. 63, No. 4, 2014, 877–888, DOI: 10.1109/TIM.2013.2283741.
15. Gibert X., Patel V.M., Chellappa R., *Deep Multitask Learning for Railway Track Inspection*, "IEEE Transactions on Intelligent Transportation Systems", Vol. 18, No. 1, 2017, 153–164, DOI: 10.1109/TITS.2016.2568758.
16. Giben X., Patel V.M., Chellappa R., *Material classification and semantic segmentation of railway track images with deep convolutional neural networks*. IEEE International Conference on Image Processing, 2015, 621–625, DOI: 10.1109/ICIP.2015.7350873.
17. Gibert X., Patel V.M., Chellappa R., *Robust fastener detection for autonomous visual railway track inspection*. IEEE Winter Conference on Application of Computer Vision, 2015, 694–701, DOI: 10.1109/WACV.2015.98.
18. He K., Gkioxari G., Dollar P., Girshick R., *Mask R-CNN*, "Computer Vision and Pattern Recognition", 2017, DOI: 10.48550/arXiv.1703.06870.
19. Karakose M., Yaman O., Murat K., Akin E., *A new approach for condition monitoring and detection of rail components and rail track in railway*, "International Journal of Computational Intelligence Systems", Vol. 11, No. 1, 2018, 830–845, DOI: 10.2991/ijcis.11.1.63.
20. Karakose M., Yaman O., Murat K., Akin E., *Real time implementation for fault diagnosis and condition monitoring approach using image processing in railway switches*. "International Journal of Applied Methods in Electronics and Computers", 2016, 307–313, DOI: 10.18100/ijamec.270627.
21. Krizhevsky A., Sutskever I., Hinton G.E., *ImageNet classification with Deep Convolutional Neural Networks*, "Advances in Neural Information Processing Systems", Vol. 14, 2012, 1097–1105.
22. Kumar B.V.K.V., Mahalanobis A., Juday R., *Correlation pattern recognition*, Cambridge University Press, 2005.
23. Lesiak P., *Mobilna diagnostyka szyn w torze kolejowym*. Monografia habilitacyjna, Wydział Transportu Politechniki Warszawskiej, 2008.
24. Lesiak P., Bojarczak P., *Application of neural classifier to railway flaw detection in the method of metal magnetic memory*, The 6 International Conference "Environmental Engineering" Selected papers, Vol. 2, 744–747, Vilnius, Lithuania, 2005.
25. Lesiak P., Bojarczak P., *Application of wavelets and fuzzy sets to detection of head – checking defects in railway rails*, "Transport Systems Telematics", 10 Conference TST 2010, Communications in Computer and Information Science, Springer, Vol. 104, 2010, 327–334, DOI: 10.1007/978-3-642-16472-9_36.
26. Lesiak P., Bojarczak P., *Inteligentne algorytmy analizy ultradźwiękowej obrazów w badaniach bezstykowych złączy szyn metodą TOFD*, „Logistyka”, 3, 2012.
27. Lesiak P., Bojarczak P., Migdal M., *Inteligentne klasyfikatory wad kontaktowo-naprężeniowych w szynach kolejowych*. „Pomiary Automatyka Komputery w Gospodarce i Ochronie Środowiska”, Nr 3, 2009, 13–17.
28. Li Y., Trinh H., Haas N., Otto C., Pankanti S., *Rail component detection, optimization, and assessment for automatic rail track inspection*, "IEEE Transactions on Intelligent Transportation Systems", Vol. 15, No. 2, 2014, 760–770, DOI: 10.1109/TITS.2013.2287155.
29. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y., Berg A.C., *SSD: Single Shot MultiBox Detector*, "Computer Vision and Pattern Recognition", 2015, DOI: 10.1007/978-3-319-46448-0_2.
30. Long J., Shelhamer E., Darrell T., *Fully Convolutional Networks for Semantic Segmentation*, "Computer Vision and Pattern Recognition", 2014, DOI: 10.48550/arXiv.1411.4038.
31. Mazzeo P., Nitti M., Stella E., Distante A., *Visual recognition of fastening bolts for railroad maintenance*, "Pattern Recognition Letters", Vol. 25, No. 6, 2004, 669–777, DOI: 10.1016/j.patrec.2004.01.008.
32. Rauschmayr N., Hochemer M., Zurkirchen M., Kenzelmann S., Gilles M., *Deep Learning of Railway Track Faults using GPUs*, Global Technology Conference, Santa Clara, USA, 2018.
33. Redmon J., Divvala S., Girshick R., Farhadi A., *You Only Look Once: Unified, Real-Time Object Detection*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016, 779–788, DOI: 10.1109/CVPR.2016.91.
34. Ren S., He K., Girshick R., Sun J., *Faster R-CNN: Towards real-time object detection with region proposal networks*, "IEEE Transactions on Pattern Recognition and Machine Intelligence", Vol. 39, 2017, 1137–1149, DOI: 10.1109/TPAMI.2016.2577031.
35. Resendiz E., Hart J., Ahuja N., *Automated visual inspection of railroad tracks*, "IEEE Transactions on Intelligent Transportation Systems", Vol. 14, No. 2, 2013, 751–760, DOI: 10.1109/TITS.2012.2236555.
36. Tan C., Sun F., Kong T., Zhang W., Yang C., Liu C., *A survey on Deep Transfer Learning*, Proceedings of 27th International Conference on Artificial Neural Networks, Rhodes, part 3, Vol. 11141, 2018, 270–279, DOI: 10.1007/978-3-030-01424-7_27.
37. Yang J., Tao W., Liu M., Zhang Y., Zang H., Zhao H., *An efficient direction field-based method for the detection of fasteners on high-speed railways*, "Sensors", Vol. 11, No. 8, 2011, 364–7381, DOI: 10.3390/s110807364.
38. Shorten C., Khoshgoftaar T.M., *A survey on Image Data Augmentation for Deep Learning*. "Journal of Big Data", Vol. 6, 2019, DOI: 10.1186/s40537-019-0197-0.

Inne źródła

39. <https://www.plastwil.pl/systemy-przytwierdzen-szyn-produkty-pl.html>, accessed 10 January 2024

Automatic Diagnosis of Railway Track Elements Using Deep Learning Neural Networks

Abstract: The article discusses a method for automatic diagnostics of a railway track. It consists in automatic evaluation of the technical condition of selected track elements, such as rails, wooden and concrete sleepers, fasteners and turnouts. It was carried out on the basis of analysis of video images of railroad track elements recorded by two line cameras placed on the diagnostic carriage. The selected FCN-8 deep learning neural network was used to assess the technical condition of the surveyed elements, and the effectiveness of the applied algorithm was determined on the basis of such measures as IoU, Precision, Recall. Conclusions on the application of the FCN-8 network in the automatic classification of features of selected railroad track elements are presented. The results obtained were compared with other methods used in vision diagnostics.

Keywords: railway track vision diagnostics, learning algorithms, image processing

dr hab. inż. Piotr Bojarczak, prof. URad

p.bojarczak@uthrad.pl

ORCID: 0000-0002-1977-9865

Absolwent Wydziału Transportu Wyższej Szkoły Inżynierskiej w Radomiu, gdzie w 1990 r. uzyskał tytuł magistra inżyniera transportu. Stopień doktora nauk technicznych w dyscyplinie elektrotechniki uzyskał w 1995 r. na Wydziale Elektrycznym Politechniki Warszawskiej, natomiast stopień doktora habilitowanego w dyscyplinie transport uzyskał w 2014 r. na Wydziale Transportu i Elektrotechniki Uniwersytetu Technologiczno-Humanistycznego w Radomiu. Obecnie pracuje w Katedrze Systemów Sterowania i Elektroniki na Wydziale Transportu, Elektrotechniki i Informatyki Uniwersytetu Radomskiego. Jego obszar zainteresowań naukowych obejmuje zagadnienia związane z zastosowaniem algorytmów uczących w szeroko rozumianej diagnostyce kolejowej.



dr hab. inż. Waldemar Nowakowski, prof. URad

w.nowakowski@uthrad.pl

ORCID: 0000-0003-0592-9065

Absolwent Wydziału Transportu Wyższej Szkoły Inżynierskiej w Radomiu. W 1988 r. uzyskał tytuł magistra inżyniera transportu, w 2009 r. stopień doktora nauk technicznych, a w 2018 r. stopień doktora habilitowanego w dyscyplinie transport. Obecnie pracuje w Katedrze Systemów Sterowania i Elektroniki na Wydziale Transportu, Elektrotechniki i Informatyki Uniwersytetu Radomskiego. Jego obszar zainteresowań naukowych obejmuje zagadnienia związane z automatyką kolejową, bezpieczeństwem i niezawodnością systemów w transporcie kolejowym, diagnostyką systemów sterowania ruchem kolejowym oraz automatyką przemysłową.

