

DETEKCE TRHLIN POMOCÍ ZPRACOVÁNÍ A ANALÝZY OBRAZU S VYUŽITÍM FUZZY LOGIKY

Ing. Václav Wudi, *

Katedra betonových a zděných konstrukcí, Fakulta stavební
České vysoké učení technické v Praze, Thákurova 7/2077, 166 29 Praha 6, Česká republika.
vaclav.wudi@fsv.cvut.cz

ABSTRAKT

Trhliny v betonu indikují překročení mezního napětí materiálu a představují klíčový faktor při hodnocení stavu betonových konstrukcí. Tento článek se zabývá vývojem algoritmu pro automatickou detekci trhlin v betonových površích na základě zpracování obrazu s využitím fuzzy logiky. Navržený algoritmus využívá fuzzy logiku se dvěma parametry pro definování funkcí příslušnosti a provádí defuzzifikaci metodou Mamdani.

Implementace byla realizována v programovacím jazyce Python a testována na sadě dvanácti obrazových vzorků betonu s různými typy trhlin i bez nich. Výsledky experimentů ukazují, že algoritmus dosahuje uspokojivé přesnosti pro většinu testovaných vzorků, přičemž identifikované nedostatky budou zohledněny v dalším výzkumu zaměřeném na jeho optimalizaci pro konkrétní aplikace.

KLÍČOVÁ SLOVA

Detekce trhlin • Obrazová analýza • Fuzzy logika • Analýza konstrukce

ABSTRACT

Cracks in concrete indicate that the material's ultimate stress has been exceeded and represent a critical factor in evaluating the condition of concrete structures. This paper presents the development of an algorithm for the automatic detection of cracks on concrete surfaces using image processing techniques based on fuzzy logic. The proposed algorithm employs fuzzy logic with two parameters to define the membership functions and applies the Mamdani method for defuzzification.

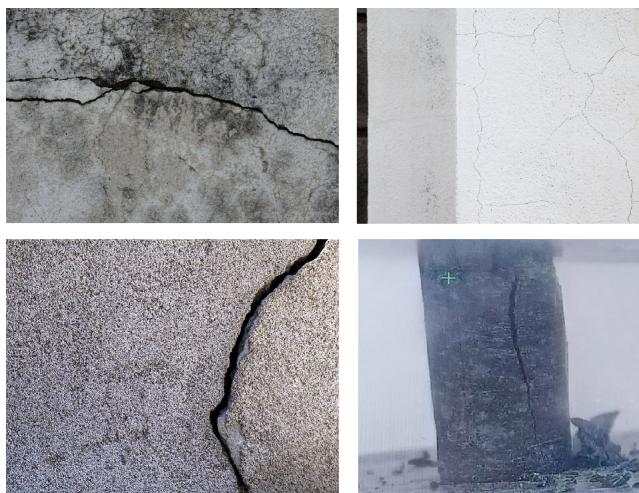
The algorithm was implemented in the Python programming language and evaluated on a dataset of twelve image samples of concrete, both with and without various types of cracks. Experimental results demonstrate that the algorithm achieves satisfactory accuracy for most of the tested samples. Identified limitations will be addressed in future research focused on optimizing the algorithm for specific applications.

KEYWORDS

Crack detection • Image processing • Fuzzy logic • Analysis of construction

1. ÚVOD

Trhliny v betonových konstrukcích, vzorcích a povrchových vrstvách betonu představují významný indikátor namáhání a mohou



Obrázek 1: Příklady trhlin: trhlina na podlaze, trhlina na betonové stěně, trhlina na betonové silnici, trhlina na zkušebním vzorku kamenniva.

vznikat při překročení mezní tahové pevnosti materiálu (viz Obr. 1).

Určení trhliny nám může říci o místě, kde dochází k překročení tahové pevnosti a tím i o vlastnostech konstrukce, jako je místo s největším napětím a největším poškozením konstrukce. Určení okraje trhliny, může říci jak veliká je trhlina, případně určit rozvoj trhlin v závislosti na času.

Pro určení poloh a velikost trhlin je vhodné použít automatické řešení. Pro řešení velkých ploch, či velký počet vzorků by bylo časově náročné řešit ručně polohu trhlin. Z polohy okraje trhlin na obraze lze vyčíst jejich velikost a umístění (viz Tian et al. (2019)).

Cílem článku bylo navrhnout jednoduchý algoritmus pro určení polohy trhlin. Součástí cílů práce bylo vytvořit sadu testovacích vzorků, na kterých se posoudí přesnost, chybovost a časová náročnost postupu.

Navržený algoritmus by měl být výchozím bodem pro další aplikace, které budou cíleny na konkrétní použití, s cílem zlepšit konkrétní parametry, které budou pro danou aplikaci potřeba.

* Supervisor: prof. Ing. Petr Štemberk, Ph.D., D.Eng.

2. ZPRACOVÁNÍ OBRAZU

Při zpracování obrazu se výpočet provádí ve dvou základních krocích. Jsou jimi předzpracování obrazu (image preprocessing) a samotné zpracování obrazu (image processing).

2.1. Image Preprocessing

Při image preprocessingu dochází k upravení obrazu, které následně umožní lepší matematické operace s obrazem. Základním prvkem pro preprocessing jsou filtry a prahování.

Při filtrech se zvýrazní, anebo zmenší některé prvky obrazu. Může se jednat o odstranění šumu, nebo zvýšení a snížení kontrastu, dále zvýšení a snížení ostroty obrazu.

Při prahování se stanoví práh. Hodnoty, které jsou nad prahem nabydou maximální hodnoty a hodnoty pod prahem nabydou nejmenší hodnoty.

Hodnota prahu může být fixně zvolená, nebo může vycházet z parametrů obrazu, jako je například Otsu prahování (viz Otsu et al. (1975)), případně proměnné prahování.

2.2. Image processing

Předmětem image processingu je zjištění potřebných dat z obrazu. Neexistuje obecný postup pro image processing. Základními kroky v image processingu je segmentace, klasifikace a interpretace (viz Chitradevi & Srimathi (2014)).

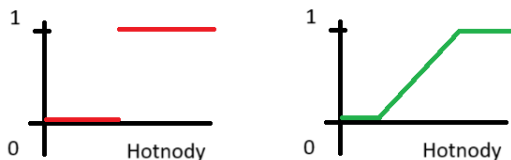
Segmentace je nalezení nějaké části. Může jít o bod, nebo linii. Jde také o ořez objektu, anebo detekci hran objektů (viz Ziou & Tabbone (1998)).

Klasifikace je nalezení nějakého vzoru. Jde o porovnání některého vzoru s objektem, nebo jeho částí. Jedním z případů klasifikace je korelace. Korelace je porovnání podobnosti dvou, nebo více stejně velkých částí obrazů.

Jednou metodou zpracování obrazu použitou v tomto článku je záplavový algoritmus. Ten nalézá spojitou plochu stejné¹ velikosti barvy pixelu (viz Levoy (1981)).

2.3. Fuzzy logika

Fuzzy logika vychází z nebinárního popisu skutečnosti, kdy pravdivost není určena jako pravda a nepravda, ale může nebývat škále hodnot v celém rozsahu od nepravdy značené 0 do pravdy značené 1 (viz Zadeh (1988), viz Obr. 2).



Obrázek 2: Pravidlost fuzzy logiky a binární logiky, vlevo binární přiřazení, vpravo pravdivost fuzzy logiky.

Nejdříve se fuzzifikují vstupní hodnoty dle fuzzy funkcí. Dále se vyhodnotí hodnoty výstupu podle všech podmínek výstupu.

Posledním krokem je defuzzifikace. Tu lze provést pomocí Mandany (viz Iancu (2012)), anebo Takagei-Sugeno (TSK) metodou.

¹Nalézá pixely stejné velikosti případně takové, které jsou v určitém intervalu. V případě aplikace v tomto článku je vstupem binární obraz a barvách 0 a 255.

Při defuzzifikaci se získá konkrétní výstup. V případě tohoto článku je to výstup: jedná se o trhlinu a nejedná se o trhlinu.

3. PARAMETRY VYHODNOCENÍ

Pro posuzování výpočtu byly vybrány tři parametry. Jsou jimi výpočtový čas, přesnost a chybovost algoritmu.

Výpočtovým časem je čas od začátku výpočtu do konce výpočtu. Uvádí se v závislosti na velikosti obrazu. Obrazy mohou mít různé velikosti a to může mít vliv na čas. Proto je lepší pracovat s časovou náročností. Ta se vyjádří jako poměr velikosti a času

$$P = \frac{A}{t} \quad (1)$$

kde P je výpočtová náročnost uváděna v pixelech za vteřinu, A je počet pixelů v obrazu, která vychází z rozlišení. Hodnota t je výpočtový čas.

Přesnost a chybovost spolu souvisí. Hranicí mezi nimi je mez přesnosti. Jakákoliv odchylka skutečnosti a predikce o velikosti do této meze je nepřesnost. Odchylka nad tuto mezi je chybovost (viz Obr. 3).



Obrázek 3: Zeleně vyznačena mez mezi přesností a chybovostí, Červeně chybné určení trhliny, modře nepřesné určení trhliny

Chybovost může být falešně pozitivní a falešně negativní, stejně tak i nepřesnost. Falešně pozitivní chyba nastává pokud algoritmus detekuje trhlinu tam, kde není. Naopak negativní chyba značí to, že algoritmus nedetekuje místo, kde je trhlina.

Chybovost a nepřesnost může nabývat rozdílných hodnot pro falešně pozitivní a negativní. Obě hodnoty jsou vyjádřeny poměrem (procenty).

Celkovou chybovost lze vyjádřit jako poměr chyb ku celkové ploše

$$E = \frac{A_e}{A} \quad (2)$$

kde E představuje chybovost, A plochu obrazu (počet celkových pixelů), případně plochu trhliny, v závislosti zda o jakou chybu se jedná. Ale představuje chybné pixely.

Nepřesnost má maximum stanovené mezní hranicí mezi chybou a nepřesností. Pro každý bod lze vyjádřit nepřesnost určením bodu. Z nich lze poté stanovit četnosti nepřesností v absolutní hodnotě, nebo poměrně. Případně lze vypočítat průměrnou nepřesnost.

4. APLIKACE

Aplikace algoritmu probíhala v programovacím jazyce Python. Výpočet je určen pro každý pixel, zda se jedná o okraj trhliny a nebo nikoliv.

4.1. Způsob výpočtu

Výpočet byl proveden pomocí programu vytvořeného v programovacím jazyce Python. Python je programovací jazyk (viz Obr. Python (2021)). Python je zvolen z důvodu množství vhodných knihoven pro image processing Guan et al. (2019))

4.2. Postup algoritmu

Algoritmus sestává ze tří hlavních částí, preprocessingu, aplikace fuzzy logiky a odstranění malých částí (viz Obr. 4)



Obrázek 4: Flowchart - postup algoritmu

Prvním krokem je preprocessing. Ten zahrnuje převedení RGB obrazu do černobílé barvy a odstranění šumu Gausovým blurem. Tím se odstraní případný bodový šum.

Následně jsou vytvořeny funkce pravdivosti pro prvky fuzzy logiky, a provedena fuzziifikace a defuzziifikace.

Posledním krokem je odstranění příliš malých částí. Jde o obdobu šumu. Pro toto odstranění určíme pomocí záplavového algoritmu spojitou oblast předpokládané trhliny.

Poté jí posoudíme vzhledem k stanovené maximální a minimální požadované ploše trhliny:

$$A_{min} < A_{cr} \quad (3)$$

$$A_{cr} < A_{max} \quad (4)$$

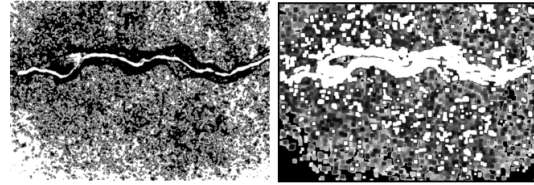
kde A_{cr} je plocha trhliny, A_{min} je minimální plocha trhliny a A_{max} maximální plocha trhliny. Aplikace kódu pro záplavový algoritmus je provedeno pomocí prohledávání do šířky (Breadth first search - BSF, viz Silvela & Portillo (2001)).

4.3. Parametry fuzzy logiky

Pro tento případ jsou použity dvě hodnotící funkce. První je hodnota pixelu (bodu) a druhou je velikost gradientu v okolí hledaného bodu (viz Obr. 5).

Vyhodnocovací podmínka pro hranici trhliny jsou určeny jako místo, kde je bod tmavý a zároveň je v jeho okolí vysoký gradient. Pro defuzziifikaci je vybrána Sugeno metoda.

Pro vyhodnocení tmavosti a světlosti jsou zvoleny dvě funkce pravdivosti. První značí pravdivost tmavosti pixelu a druhá světlosti pixelů.



Obrázek 5: Parametry fuzzy logiky vyznačené černobíle, vlevo je barevnosti bodů, vpravo maximální gradient v okolí bodu.

Pro vyhodnocení velikosti gradientu je třeba nejdříve vytvořit gradient v každém bodě. Ten vychází z rozdílů velikosti hodnot pixelů v X a Y směru. Poté se znormalizuje tak, aby maximální činil maximální hodnotu obrazu (255).

Pro každý pixel byl poté nalezen maximální gradient ve vzdálenosti do sedmi pixelů. Funkce příslušnosti gradientu je popsána dvěma funkcemi. Jsou jimi nízký a vysoký gradient.

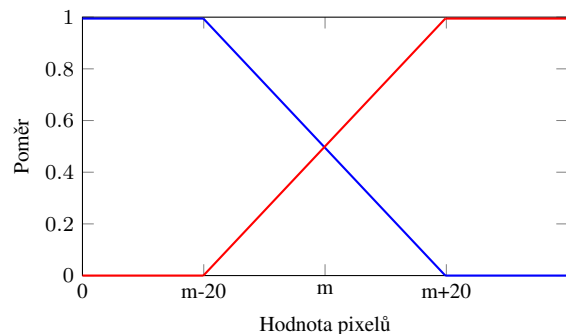
4.4. Vstupní parametry

Základní vstup jsou testovací vzorky. Kromě nich jsou zde popsány parametry, které byly zvoleny ve výpočtu. Oblast, která se zahrnuje do rozmezí hrany trhliny je 7 pixelů, stejně jako porovnávací hranice pro chybovost.

Pro aplikaci Gausovým blurem je použit filtr o velikosti 5x5. Pro odstranění šumu na konci výpočtu je použito okolí bodu o velikosti 4 a 30 procentní minimální zaplnění.

Rozmezí pro funkce pravdivosti fuzzy logiky vychází z vlastností obrazu, mediánu. Jejich zlomy jsou 20 pixelů pod a 20 nad mediánem (viz Obr. 6).

Minimální velikost spojitě plochy pro trhlínu je stanoveno na 200 pixelů, maximální jako 5 % plochy obrazu.



Obrázek 6: Funkce příslušnosti fuzzy logiky.

4.5. Testovací vzorky

Pro otestování vhodnosti algoritmu vzhledem k parametrům bylo vybráno dvanáct fotografií betonových ploch jako testovací sada vzorků. Jsou to fotografie, které reprezentují různé trhliny a také fotografie betonu bez trhlín.

Celkem se jedná o sadu dvanácti testovacích obrazů. Různé typy trhlín jsou zastoupeny ve skupinách vždy o třech fotografiích. Jedná se o čtyři skupiny.

První skupina fotografií reprezentuje rozvětvené trhliny na poškozeném vzorku s proměnnou barevností okolí trhlín. Druhou skupinou jsou obecné široké trhliny. Třetí skupinou jsou tenké trhliny

na světlém pozadí pozadí. Posledními třemi vzorky jsou obrazy betonu bez trhliny. U nich jsou vybrány takové vzorky, kde by mohlo docházet k falešné detekci trhlin. Povrch má u nich proměnou světlou, nebo hrubou strukturu. (viz Obr. 7).



Obrázek 7: Příklad testovacích vzorků, každý řádek představuje skupinu vzorků

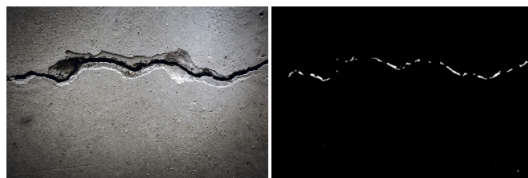
Algoritmus byl vytvořen při průběžné kontrole a vyhodnocení na jednom obraze a poté byly testovací vzorky vybrány. Na vzorcích byly nalezeny trhliny nejdříve manuálně a poté aplikován algoritmus.

5. VÝSLEDKY

V této kapitole jsou představeny výsledky. Algoritmus byl aplikován na testovací vzorky. Jsou jimi grafické výstupy fotografií a tři parametry vyhodnocení.

5.1. Obrazové výsledky

Prvním výsledkem je testovací obraz. Ten zahrnuje detekovaný výsledek trénovacího vzorku (viz Obr. 8).



Obrázek 8: Příklad výsledku, vlevo fotografie trhliny a vpravo je detekce, bíle předpokládaná trhlina.

Dalšími obrazovými výsledky jsou výstupy měřené sady vzorků (viz kapitola Přílohy).

5.2. Parametry

Při testování byly měřeny parametry vzorků. Jsou jimi výpočtový čas, chybovost a přesnost.

První z nich je výpočtový čas. Výsledky času jsou vyjádřeny jako časová efektivita v kilo-Pixelech za sekundu (viz Tab. 1). Celková průměrná efektivita výpočtového času byla 16.1 kilo-pixelů za sekundu.

Tabulka 1: Efektivita výpočtového času, vyjádřeno kilo-Pixelech za sekundu

Vzorek	Sada 1	Sada 2	Sada 3	Sada 4
1	16.1	15	19	15,1
2	16	16.6	18	14.2
3	16.8	15.8	15.9	15.6
Σ	16.3	15.8	17.6	15

Dalším měřeným parametrem je chybovost. Ta je rozdělena na falešně pozitivní a falešně negativní. První je uvedena falešně pozitivní (viz Tab. 2). Průměrná falešně pozitivní chybovost je 16.2 %.

Tabulka 2: Falešně pozitivní chybovost testovacích vzorků uvedená v procentech

Vzorek	Sada 1	Sada 2	Sada 3	Sada 4
1	0.6	2.3	0.1	3.7
2	6.8	0.3	0	4.5
3	0.3	0.4	1.3	4
Σ	2.5	1	0.5	4.1

Dalším výsledkem je falešně negativní chybovost. U falešně negativní nemá smysl uvádět hodnoty pro obrazy bez trhlin, protože jsou z definice nulové (viz Tab. 3). Průměrná falešně negativní chybovost je 2.7 %.

Tabulka 3: Falešně pozitivní chybovost testovacích vzorků uvedená v procentech

Vzorek	Sada 1	Sada 2	Sada 3
1	0.2	0	0
2	19.3	20.7	100
3	26.3	0	1.4
Σ	15.3	6.9	33.8

Dalším parametrem je přesnost. Ta je vyjádřena pro každý bod trhliny, jako vzdálenost nejbližšího bodu, kde došlo k detekci (viz Obr. 9)².

6. DISKUZE

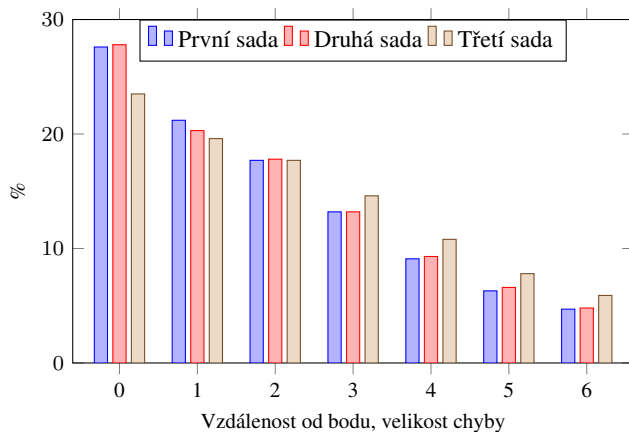
Algoritmus vykazuje malou míru chybovosti u vzorků, kdy je vysoký kontrast mezi trhlinou o okolím trhliny a zároveň když je trhlina dostatečně veliká.

Pro jeden vzorek došlo k nenalezení trhliny. Jedná se o druhý vzorek z třetí sady. Jde o 100 procentní falešně negativní chybovost. U tohoto vzorku je beton mimo trhlinu rozdílný a má výrazné přechody barev. Při zmírnění podmínek odstraňování šumu lze docílit alespoň částečné detekce. U jiného vzorku se nepovedlo zachytit jednu tenčí větev rozvětvené trhliny.

U vzorků bez trhlin došlo k nesprávné detekci a tím falešně pozitivní chybě. Vzorky byly schválně vybrány tak, aby měli proměnnou barevnost, případně různé nepravidelnosti odstínu.

Nevýhodou algoritmu může být jeho časová náročnost, zvláště při aplikaci na velké plochy. To vyplývá s požadavků, kdy se při návrhu algoritmu nebral zřetel na časovou náročnost. Vysoká časová náročnost vychází jednak ze samotné podstaty fuzzy logiky a zároveň hlavně z důvodu aplikace výpočtu pro každý pixel.

² Ve vyjádření nepřesnosti není zahrnut jeden vzorek u kterého nedošlo k detekci trhliny. Nepřesnost by vycházela matematicky nulově. V třetí sadě se průměrují jen zbylé dva vzorky



Obrázek 9: Graf distribuce nepřesností, výsledky průměrem pro každou první tři sady vzorků.

Časová náročnost 16 kilo-pixelů za vteřinu odpovídá zpracování 2 fullHD fotografií za minutu. To může být dostatečné pro fotografie v řádech jednotek, případně desítek kusů.

7. ZÁVĚR

Byl vytvořen algoritmus pro detekci trhliny pomocí image processingu, který vykazuje malou chybovost u vzorků, kde je jasná trhlina. Algoritmus by při správném osvětlení mohl být použit pro detekci nejvíce porušených míst vybrané konstrukce.

Další směr vývoje závisí na požadavcích aplikace. Jedním z možností rozvoje je zvýšit rychlosti výpočtu pro aplikace velkých ploch, jako jsou vozovky a povrchy budov.

Druhým směrem, kterým je třeba se zabývat, je zvýšit přesnost u malých trhlin a u trhlin celkově pro detekci trhlin u měřených vzorků.

Poděkování

Tento článek vznikl za finanční podpory ČVUT v Praze v rámci projektu SGS25/039/OHK1/1T/11.

Reference

- Chitradevi, B. & Srimathi, P. (2014), 'An overview on image processing techniques', *International Journal of Innovative Research in Computer and Communication Engineering* 2(11), 6466–6472.
- Guan, Y., Zhou, F. & Zhou, J. (2019), Research and practice of image processing based on python, in 'Journal of Physics: Conference Series', Vol. 1345, IOP Publishing, p. 022018.
- Iancu, I. (2012), 'A mamdani type fuzzy logic controller', *Fuzzy logic-controls, concepts, theories and applications* 15(2), 325–350.
- Levoy, M. (1981), 'Area flooding algorithms', *Two-Dimensional Computer Animation, Course Notes 9 for SIGGRAPH* 82.
- Otsu, N. et al. (1975), 'A threshold selection method from gray-level histograms', *Automatica* 11(285-296), 23–27.
- Python, W. (2021), 'Python', *Python Releases for Windows* 24.

Silvela, J. & Portillo, J. (2001), 'Breadth-first search and its application to image processing problems', *IEEE Transactions on Image Processing* 10(8), 1194–1199.

Tian, F., Zhao, Y., Che, X., Zhao, Y. & Xin, D. (2019), 'Concrete crack identification and image mosaic based on image processing', *Applied Sciences* 9(22), 4826.

Zadeh, L. A. (1988), 'Fuzzy logic', *Computer* 21(4), 83–93.

Ziou, D. & Tabbone, S. (1998), 'Edge detection techniques-an overview', *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications* 8(4), 537–559.

Fotografie

Akyurt E. (2022), Black and white photo of a concrete wall, Dostupné z: <https://unsplash.com/photos/a-black-and-white-photo-of-a-concrete-wall-P4Z6mCRdMp0lacht>

Mockaroon (2018) Marble Hotel Wall, dostupné z: <https://unsplash.com/photos/a-black-and-white-photo-of-a-wall-YqUeLG7fMr4>

Crack concrete, <https://perfectpowerwash.com/blog/why-concrete-cracks-how-to-protect-it/>

freepig, Wall close up dostupné z: https://www.freepik.com/free-photo/wall-close-up_4089681.htm#fromView=search&page=1&position=36&uuiid=ff0b3a89-1eb0-480b-b8c3-5931d973d676&query=concrete+floor+crack

Froghugger, Uzavřete fotografii trhliny v betonové podlaze. Konstrukční opravy a konstrukce, dostupné z: <https://depositphotos.com/cz/photo/close-photo-crack-concrete-floor-structural-repair-construction-375777806.html>

wirestock https://www.freepik.com/free-photo/closeup-shot-crack-concrete-wall_8048636.htm#fromView=search&page=1&position=22&uuiid=acca6493-af04-41e2-9ae6-52a479394efe&query=concrete+road+crack

Přílohy

Výsledky detekce obrazů, řádek představuje sadu vzorků, bílou barvou je detekovaná trhlina, černou je vyznačena nedetekovaná oblast.

