

» Razpoznavna oblik na principu strojnega učenja

Jožica Piškur
Erih Arko
Uroš Kromar
Marko Munih

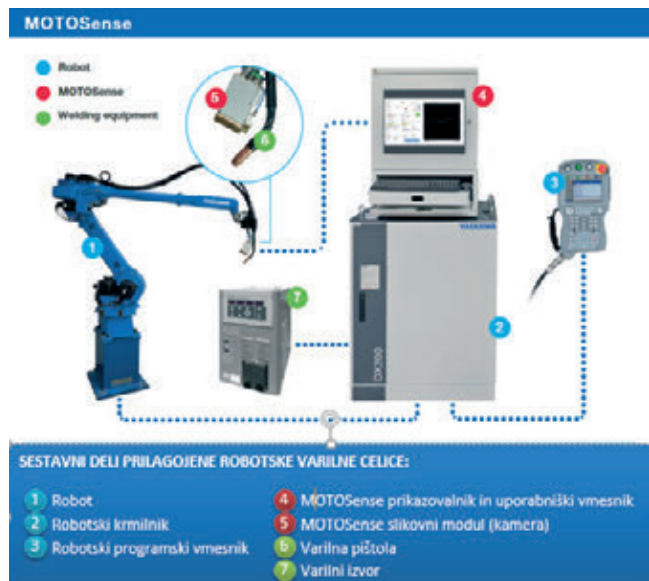
Princip vodenja robota s pomočjo strojnega vida je, da na podlagi oblike zanimanja načrtamo trajektorijo. Če obliko zanimanja izgubimo, s tem izgubimo tudi trajektorijo. Varilna robotska aplikacija s sistemom MOTOSense omogoča omenjeno vodenje robota pri varjenju (Slika 1), pri čemer robot sledi sočelnemu spoju dveh kovin. Kovini sta predhodno zvarjeni s točkovnimi zvari, ki nam predstavljajo oviro na trajektoriji. Zaradi teh zvarov lahko robot zavije z želene trajektorije ter tako kovini nista pravilno zvarjeni.

Zato smo v okviru seminarja raziskali trenutno stanje na področju strojnega vida na omenjeno problematiko. Pri tem smo analizirali obstoječa programska okolja, ki bi bila sposobna prepoznati točkovne zware ter podati informacijo, kje se nahajajo. Osredotočili smo se na programsko okolje ViDi Suite [1], ki izkorišča teorijo globokega učenja, ter na programsko knjižnico OpenCV [2].

MOTOSense Vision system

Z razvitim sistemom za strojni vid MOTOSense omogočamo adaptivno korekcijo premika robota, čigar natančnost znaša 0,1 mm med samim tehnološkim procesom. Je programsko prilagodljiv za razpoznavo najbolj kompleksnih oblik objekta. Uporablja se v tehnoloških procesih, kjer se pozicija obdelovalne površine

spreminja in ni možna uporaba dragih pozicionirnih orodij. Najpogostejše je to v splošni industriji. Kot končne izdelke pa predstavljajo gradbeni in kmetijski stroji, cisterne in izdelki za živilsko industrijo [3].



» Slika 1. MOTOSense zgradba



» Slika 2. Kovinska cisterna s točkovnimi zvari



» Slika 3. Varjenje s pomočjo sistema MOTOSense

Razpoznavna točkovnih zvarov

Pri varilnih robotskih aplikacijah brez kamere robot naučeno vari kovini po natančno določeni trajektoriji, kjer je postavitev kovin natančno določena. Sistem MOTOSense s skenirno napravo in kamero na vrhu robota omogoča varilni aplikaciji adaptivno korekcijo premika robota med samim varjenjem (Slika 4). Slikovna obdelava videa iz kamere poteka v programskem okolju LabVIEW, ki omogoča blokovno programiranje.

Velikokrat pa sta kovini že predhodno točkovno zvarjeni (Slika 2), da je lažja postavitev kovin na delovno podlago. To pa predstavlja problem pri sledenju spoja, saj ga točkovni zvari prekinjajo. Posledično robot pri točkovnem zvaru pogosto zavije s poti oz. trajektorije (Slika 3).

Da bi robot nadaljeval trajektorijo kljub motnjam, tj. točkovnim zvarom, mora le-te prepoznati, ter ob nastopu točkovnega zvara in-



Jožica Piškur, Marko Munih - Fakulteta za elektrotehniko, Univerza v Ljubljani
Erih Arko, Uroš Kromar - Yaskawa Ristro, d. o. o.

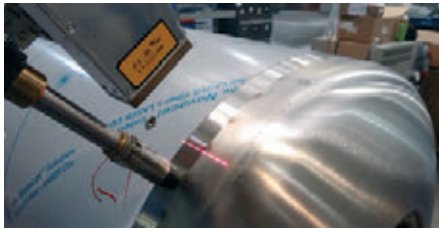
The KUKA logo is displayed in a bold, orange, sans-serif font in the top left corner. The background of the entire page is a complex, abstract digital network of grey and white lines and polygons, with a central glowing orange and white circular element that resembles a stylized sun or a data hub.

Hello Industrie 4.0 _we connect you

Internet stvari, Industrija 4.0 ali pametna proizvodnja – čeprav se uporabljajo različna poimenovanja, je osnovna zamisel vedno enaka. Gre za dolgoročno transformacijo našega globalnega dojemanja industrijske proizvodnje s povezovanjem digitalnega in fizičnega sveta v eno celoto. KUKA ima kot vodilna avtoriteta in pionir na področju Industrije 4.0 odločilno vlogo pri prehodu v omreženo, pametno proizvodnjo. S pogumom, da spreminjamo stvari.

www.kuka.com

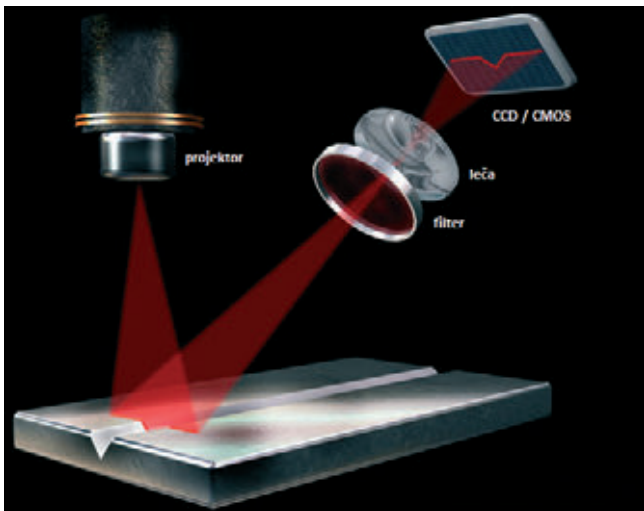
terpolirati trajektorijo [4], [5]. Za prepoznavanje zvarov smo tako analizirali programsko okolje ViDi Suite ter programsko knjižnico OpenCV.



» Slika 4. Skeniranje kovine po načelu triangulacije

Skenirna naprava

Skenirna naprava deluje po načelu laserske triangulacije (Slika 5). Kovino, ki jo skeniramo, se osvetli z laserskim žarkom, ki po kovini potuje linijsko. Kamera spremlja svetlobo, ki se odbije od kovine. Nato analiziramo odboj z namenom določitve razdalje med senzorjem in osvetljeno površino. Ta postopek imenujemo triangulacijski algoritem.



» Slika 5. Princip triangulacije

Razpoznava s programskim okoljem ViDi Suite

ViDi Suite je programsko okolje podjetja ViDi System, ki je osnovano na algoritmih strojnega vida oz. globokega učenja. Namenjeno je za prepoznavanje, označevanje ter klasificiranje vzorcev. Sestavljajo ga tri orodja [6], ki so:

- ViDi blue,
- ViDi red,
- ViDi green.

ViDi blue je namenjen za prepoznavanje vzorcev ter njihovo identifikacijo. Pri tem orodju najprej zberemo slike vzorcev, ki jih želimo prepoznati (točkovni zvari). Le-te identificiramo ter tako naučimo model. Sledi testiranje modela in potrjevanje prepoznavanja vzorcev na nenaučenih slikah.

ViDi red deluje na dva načina: kot detektor anomalij, kjer učimo orodje dobrih vzorcev; ter kot detektor napak, kjer učimo orodje slabih vzorcev. Pri prvem načinu zberemo slike dobrih vzorcev, s katerimi s pomočjo ViDi red naučimo model. Le-tega preizkusimo na slikah s slabimi vzorci. Pri drugem načinu pa zberemo slike tako dobrih (brez točkovnih zvarov) kot slabih vzorcev (s točkovnimi zvari). Pri tem posebej označimo tako dobre kot slabe vzorce na slikah. Nato s pomočjo dveh ločenih skupin vzorcev naučimo model. Model učimo z orodjem ViDi red supervised. Sledi preizkušanje zgrajenega modela na nenaučenih slikah.

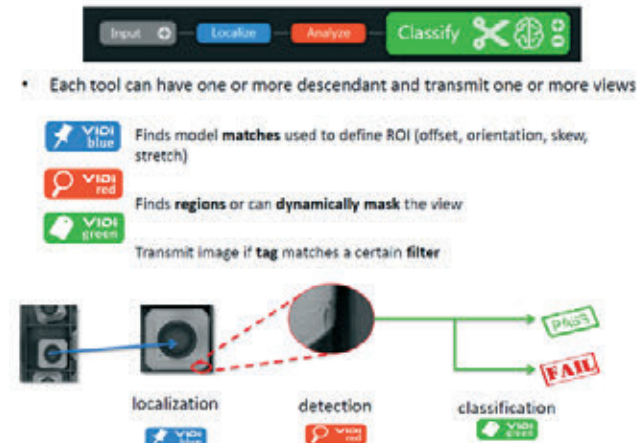
ViDi green je orodje, ki je namenjeno za označevanje prizorov ter klasifikacijo vzorcev, ki so v našem primeru točkovni zvari. Zopet

zberemo slike z vzorci, ki pa jim dodamo oznako, npr. točkovni zvar, spoj. Model s pomočjo orodja ViDi green naučimo prepoznane vzorce skupaj z oznakami. Sledi preizkušanje modela na nenaučenih slikah.

Pri učenju modela smo upoštevali naslednje parametre [6]:

- vzorčni parametri,
- učni parametri,
- parametri motenj,
- parametri procesiranja.

Z vzorčnim parametrom vplivamo na velikost značilke, kakšne barvne kanale uporabljamo ter na mejni tip slike. Z velikostjo značilke nastavimo velikost vsakega obdelanega vzorca. Barvni kanal



» Slika 6. Orodja ViDi Suite-a

slik lahko nastavimo na sivinske, spektralne ali BGR. Pri mejnem tipu slike pa nastavimo, kakšne vrednosti slikovnih elementov želimo imeti v razširjenem delu slike.

Pri učnih parametrih nastavljam dobo štetja ter izbiro učenja. Doba štetja predstavlja število optimizacijskih iteracij opravljenih v učenju. Z izbiro učenja pa naključno izbiramo odstotek slik.

Da povečamo število vzorcev in da s tem pridobimo boljše prepoznavanje vzorcev, uporabimo parameter motenj. Le-temu nastavimo, na kakšen način naj poveča število vzorcev. Npr., da vzorce v določenem intervalu skalira in/ali zrcali in/ali rotira.

Za precizno nastavljanje vzorčne gostote slike, kjer želimo vzorčiti na večji ali na manjši interval, uporabljamo parameter procesiranja.

Razpoznava s programsko knjižnico OpenCV

OpenCV je programska knjižnica, ki je osnovana na algoritmih strojnega vida. Podprta je za programske jezike, kot so C, C++, JavaScript ter Python. Raziskavo in analizo smo opravili v programskem jeziku Python v programskem okolju Spyder.

Pri tem smo uporabili algoritem ujemanja predlog (angl. Template Matching), kjer na vhodni sliki poiščemo lokacijo podlage, ki se nahaja na sliki. Z matematičnega vidika se pri omenjenem algoritmu izvaja konvolucija. Ta za vsak slikovni element izračuna odstotek prekrivanja iskane podlage v sliki. Ujemanje predlog je izvedljiva s šestimi različnimi primerjalnimi metodami [7], [8]:

1. s kvadratno razliko (1)

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

2. s kvadratno normirano razliko (2)

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I'(x + x', y + y'))$$

3. s križno korelacijo (3)

$$R(x, y) = \frac{\sum_{x', y'} T'(x', y') \cdot I'(x + x', y + y')}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}}$$

4. z normirano križno korelacijo (4)

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sum_{x', y'} T(x', y')^2 + \sum_{x', y'} I(x + x', y + y')^2}$$

5. s korelacijskim koeficientom (5)

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

6. z normiranim korelacijskim koeficientom (6)

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 + \sum_{x', y'} I(x + x', y + y')^2}}$$

Rezultati razpoznavanja točkovnih zvarov in razprava

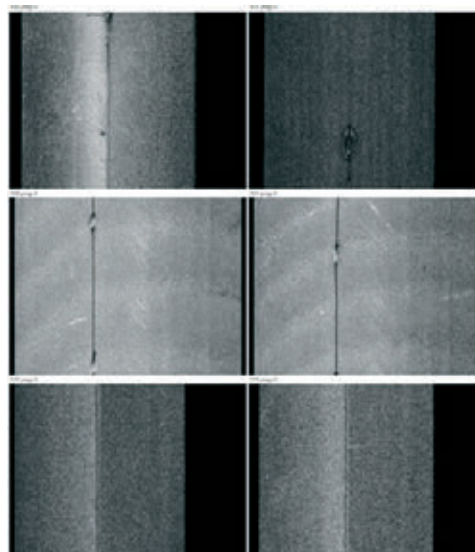
ViDi Suite

Ker ima programsko okolje ViDi Suite vprašljivo donosnost investicije, saj testna verzija z vsemi orodji stane 3.150 evrov, smo se povezali s strokovnjaki iz ViDi Systema. Z njimi smo s pomočjo orodja ViDi red supervised učili prepoznavati vzorce spoja brez ter s točkovnimi zvari iz 20 slik. S pomočjo prepoznanih vzorcev smo naučili model. Naučeni model smo nato uporabili še na preostalih 127 slikah (Slika 7).

Pri tem se je naučeni model izkazal za dobrega, saj je točkovne zveze dobro prepoznal na slikah, kjer so bili zvari izraziti (Slika 7 in Slika 8, podsliki a in b). Na slikah, kjer so bili točkovni zvari manj izraziti (Slika 7 in Slika 8, podsliki c in d), opazimo, da je model

zveze tudi prepoznal. Vendar je pri tem prepoznal tudi nepravilnosti na kovini. Npr. na sliki 8 na podsliki c se opazi manjše območje ob spoju, ki je ravno tako označeno. Na slikah, kjer točkovnih zvarov ni bilo, pa je te prepoznal, kot da so dobri vzorci.

Slika 9 prikazuje mejo med značilkami dobrih in slabih vzorcev.



» Slika 7.
Posnetki kovine
s točkovnimi
zvari

Meja oz. prag pove, s kolikšno verjetnostjo vsebuje podani posnetek dobre oz. slabe vzorce. Opazimo, da se meja dobrih vzorcev, tj. sočelni spoj brez točkovnih zvarov, nahaja med pragoma 0 do 0,31. Medtem se prag za slabe vzorce, ki jih predstavljajo točkovni zvari, nahaja med 0,32 do 1.

Skladiščna tehnika za obdelovalce kovin

OHRA 
SKLADIŠČNI SISTEMI S KONCEPTOM



Za večjo varnost v skladišču – masivni regalni sistem iz toplo valjanih kovinskih profilov

Ugodne rešitve skladiščenja za vaše kovinske izdelke

- Konzolni regali
- Konzolni regali s poševnimi konzolami za rokovanje s konzolnim dvigalom
- Paletni regali
- Kontinuirano skladiščenje dolgega blaga
- Avtomatski in polavtomatski skladiščni sistemi
- Podesti

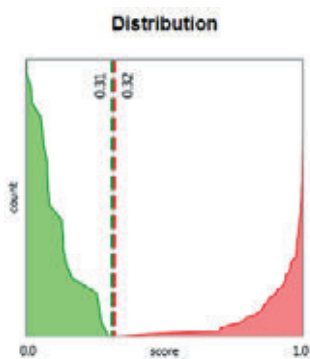
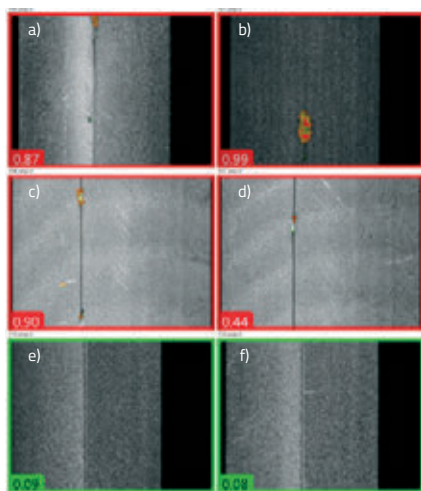
OHRA Regalanlagen GmbH
Alfred-Nobel-Str. 24 - 44
50169 Kerpen (obok Köln)

Zastopstvo in svetovanje za Slovenijo in Hrvaško:
Tel: +386 / 40 825 699 e-mail: kocjanc@ohra.de

OHRA 
SKLADIŠČNI SISTEMI S KONCEPTOM

Omenjeni model bi se lahko še izboljšal, če bi pri učenju modela uporabili večjo množico slik. To bi dosegli tudi tako, da povečamo parameter motenj. Poleg tega bi lahko povečali parameter procesiranja ter število optimizacijskih iteracij v učenju, da čim bolj izločimo nepravilnosti [9].

» Slika 8. Posnetki kovin po obdelavi z ViDi red



» Slika 9. Distribucija, ki prikazuje mejo oz. prag med značilkami dobrih in slabih vzorcev

OpenCV

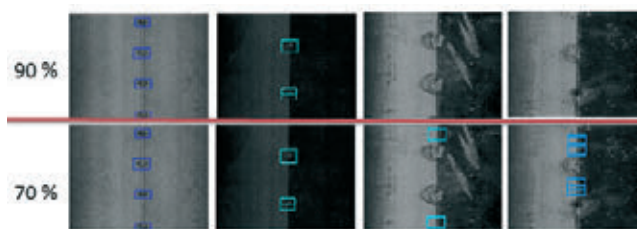
Za vhodne slike smo vzeli 39 slik dveh različnih kovin iz iste baze slik, kakor pri ViDi Suite. Za podlage smo iz 16 testnih slik izluščili podslike točkovnih zvarov. Te smo dodatno pošumili, da smo razširili bazo podslik na 630 podslik oz. podlag.

Čas procesiranja ene slike z vsemi podlagami z omenjenim algoritmom je znašal povprečno 3 s, skupno do 40 min. Algoritem je pri primerjalni metodi s korelacijskim koeficientom (enačba 6) dobro prepoznal točkovne zve iz testnih posnetkov tako pri 90 % kot pri 70 % podobnosti. Na posnetkih, kjer točkovnih zvarov nismo izluščili, le-ti pri 90 % podobnosti ni prepoznal. Pri 70 % podobnosti pa je le-te narobe prepoznal - kot je razvidno s Slike 10, spodnji desni sliki.

Ker podani rezultati niso zadovoljivi, smo problem naprej reševali tako, da smo podlage razdelili glede na tip kovine ter jih za ta namen še dodali. Podlage smo ločili na nerjavno in bakreno pločevino, oz. glede na obliko točkovnega zvara. Posnetke in podlage smo dimenzijsko zmanjšali z namenom krajšega časa procesiranja. Pri metodi s korelacijskim koeficientom se je stanje ujemanja izboljšalo. Še vedno pa so se pojavila odstopanja pri nenaučenih posnetkih.

Zato smo še preizkusili primerjalni metodi s kvadratno razliko ter kvadratno normirano razliko. Žal pa omenjeni metodi nista dajali ustreznih rezultatov. Razlog bi lahko bil v programski kodi, ki je bila enaka kot pri prejšnji metodi, ali pa bi se morali še dodatno poglobiti v teorijo primerjalnih metod [8].

Nadaljnje možne izboljšave bi bile, da povečamo število podlag ter testiramo algoritem v programskem jeziku C++. H krajšemu



» Slika 10. Prepoznavanje točkovnih zvarov z algoritmom ujemanja predlog pri 90 % in 70 % podobnosti

času procesiranja bi pripomoglo tudi procesiranje na grafični enoti. Razpoznavanje bi lahko izboljšali tudi z dodatno knjižnico Theano, ki je namenjena za uporabo algoritmov iz globokega učenja.

Zaključek

Možnosti za razpoznavo točkovnih zvarov je veliko. Zato smo se osredotočili samo na programsko okolje ViDi Suite ter na programsko knjižnico OpenCV in ju analizirali. ViDi Suite bi bil glede na rezultate dobra rešitev za naš primer, vendar ga omejuje vprašljiva donosnost investicije. OpenCV s pomočjo algoritma ujemanja podlag poda rezultate, ki kažejo na rešitev, vendar bi bilo potrebno imeti za analizo še več časa. Razlog je v dolgem času procesiranja ter v potrebi po testiranju in raziskavi drugih možnosti algoritma in knjižnice. Dodatna izpopolnitev sistema se vidi tudi v novejši veji strojnega vida, v t. i. globokem učenju [10]. Pri tem pa si moramo vzeti več časa za osvojitve potrebnih znanj.

Zahvala

Zahvaljujemo se prof. dr. Marku Munihu, iz Laboratorija za robotiko, Erihu Arku ter Urošu Kromarju, iz podjetja Yaskawa Ristro, d. o. o., za pomoč in usmerjanje v času projekta. Zahvaljujemo se tudi Petru Arku, iz podjetja Yaskawa Ristro, d. o. o., za poslane slike točkovnih zvarov ter za dodatno pomoč.

Literatura

- [1] „ViDi | Deep learning | Image Analysis Software - ViDi Systems - Vision Software“. [Na spletu]. Dostopno: <https://www.vidi-systems.com/>. [Dostopano: 20-dec-2016].
- [2] „OpenCV“. [Na spletu]. Dostopno: <http://opencv.org/>. [Dostopano: 20-dec-2016].
- [3] „MOTOSense: Machine vision system for adaptive robotic welding“, YASKAWA. [Na spletu]. Dostopno: https://www.yaskawa.eu.com/fileadmin/Download_Center/Brochures_Catalogues/Robotics/Equipment/Flyer_Equipment_MotoSense_E_06.2016.pdf.
- [4] S. B. Chen, Y. Zhang, T. Qiu, in T. Lin, „Robotic welding systems with vision-sensing and self-learning neuron control of arc welding dynamic process“, J. Intell. Robot. Syst., let. 36, št. 2, str. 191–208, feb. 2003.
- [5] J. Guenther, P. M. Pilarski, G. Helfrich, H. Shen, in K. Diepold, „Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning“, Mechatronics, let. 34, str. 1–11, mar. 2016.
- [6] „ViDi_Industry-Overview_Print.pdf“.
- [7] „OpenCV: Template Matching“, http://docs.opencv.org/3.1.0/de/da9/tutorial_template_matching.html. [Na spletu]. Dostopno: http://docs.opencv.org/3.1.0/de/da9/tutorial_template_matching.html. [Dostopano: 22-dec-2016].
- [8] „Template Matching Algorithm - OpenCV Q&A Forum“. [Na spletu]. Dostopno: <http://answers.opencv.org/question/5655/template-matching-algorithm/>. [Dostopano: 22-dec-2016].
- [9] E. N. Malamas, E. G. M. Petrakis, M. Zervakis, L. Petit, in J. D. Legat, „A survey on industrial vision systems, applications and tools“, Image Vis. Comput., let. 21, št. 2, str. 171–188, feb. 2003.
- [10] J. Schmidhuber, „Deep learning in neural networks: An overview“, Neural Netw., let. 61, str. 85–117, jan. 2015.