

Aufgabenstellung für eine Abschlussarbeit

Konfigurations- und Administrationskonzepte für Smart MMIR

Eine Erweiterung und Anwendung des GMAF Frameworks.

Lehrgebiet Multimedia und Internetanwendungen

Prof. Dr.-Ing. Matthias Hemmje

Fernuniversität Hagen

Universitätsstrasse 1, D-58097 Hagen

Raum: 1 F09, Gebäude 3 (Informatikzentrum, Universitätsstr. 1)

Tel.: +49 2331 987-304

Fax: +49 2331 987-4487

Internet: <http://www.fernuni-hagen.de>

Bearbeitungszeitraum: ab Juli 2022

Betreuer: Dipl. Inform. (Univ.) Stefan Wagenpfeil

E-Mail: stefan.wagenpfeil@fernuni-hagen.de

Skype: stefan_wagenpfeil

Abstract: Smart Multimedia Information Retrieval (MMIR) zeichnet sich durch hohe Merkmalsintegration verschiedener Medien, Skalierbarkeit und Erklärbarkeit von MMIR-Prozessen und -Objekten aus. Diese "smarten" Bestandteile eröffnen neue und zusätzliche Möglichkeiten für die Präsentation der Inhalte und die Interaktionsmöglichkeiten des Benutzers mit Smart MMIR Systemen. Im Rahmen dieser Abschlussarbeit werden Konzepte entwickelt und implementiert, die Administratoren die Konfiguration von Smart MMIR ermöglichen. Die Basis stellt hierfür das Generic Multimedia Analysis Framework (GMAF) dar.

Keywords: GMAF, MMFG, Benutzerinteraktion, MMIR

Lehrgebiet Multimedia und Internetanwendungen:

- Daten- und Dokumentenmanagement
- Informations- und Wissensmanagement
- Multimedia
- Informationssysteme und Datenbanken
- Informationsvisualisierung
- Information Retrieval
- Visuelle
- Mensch-Maschine-Interaktion
- Content- und Wissensmanagement im Internet
- Semantic Web
- Digitale Langzeitarchivierung
- Virtuelle Forschungsumgebungen
- Big Data Analyse
- Analyse natürlicher Sprache
- Berufliche Weiterbildung und E-Learning
- Industrie 4.0 und "Factories of the Future"

Die nachfolgenden Beschreibungen für die Abschlussarbeit im Bereich Informatik benennen die nötigen Aufgaben und Teilaufgaben und sollen ebenfalls als Übersicht und Leitfaden für die zu erbringenden Leistungen dienen. Zur Bearbeitung der nachfolgenden Aufgaben werden folgende Fähigkeiten und Kenntnisse vorausgesetzt:

- **Wissenschaftliches Arbeiten:** strukturiertes, analytisches Vorgehen auf Basis der gültigen Konventionen für Aufbau, Zitierweise und inhaltliche Bearbeitung von forschungsrelevanten Themen sind die Voraussetzung für das Schreiben einer Bachelor- oder Master-These. Hierzu gehören die eigenständige Literaturrecherche, die selbstständige Aneignung von Wissen und die Umsetzung von theoretischen Konzepten in die Praxis.
- **Programmiersprachen / Beschreibungssprachen:** Java, XML. Von Vorteil sind Kenntnisse im Umgang mit Eclipse und ein grundlegendes Verständnis von Ontologien und RDFs.
- **Sonstige Systeme oder Konzepte:** Aufbau und Handhabung von Graph-Strukturen, Objektorientierung, grundlegende Konzepte der semantischen Modellierung, RDF, Semantic Web.

1. Themenbeschreibung

Smart Multimedia Information Retrieval (MMIR) führt eine Reihe von Konzepten ein, die über klassisches Information Retrieval hinaus gehen. Hierzu gehören erweiterte Metriken für die Berechnung von Ähnlichkeit oder Empfehlungen, Möglichkeiten der Merkmalsintegration und -fusion, sowie Erklärbarkeit von MMIR Prozessen und Inhalten. Unter Erklärbarkeit versteht man in diesem Zusammenhang menschen-verständliche Antworten auf Fragestellungen wie "warum ist das Element in der Ergebnisliste?", oder "warum ist das Element vor dem anderen?". Auch können Merkmale und deren Eigenschaften menschen-verständlich erklärt werden und somit zum Schließen der Semantischen Lücke beitragen. Die Eigenschaften von Smart MMIR werden im Generic Multimedia Analysis Framework (GMAF) vereint, welches mittels einer API verwendet und integriert werden kann.

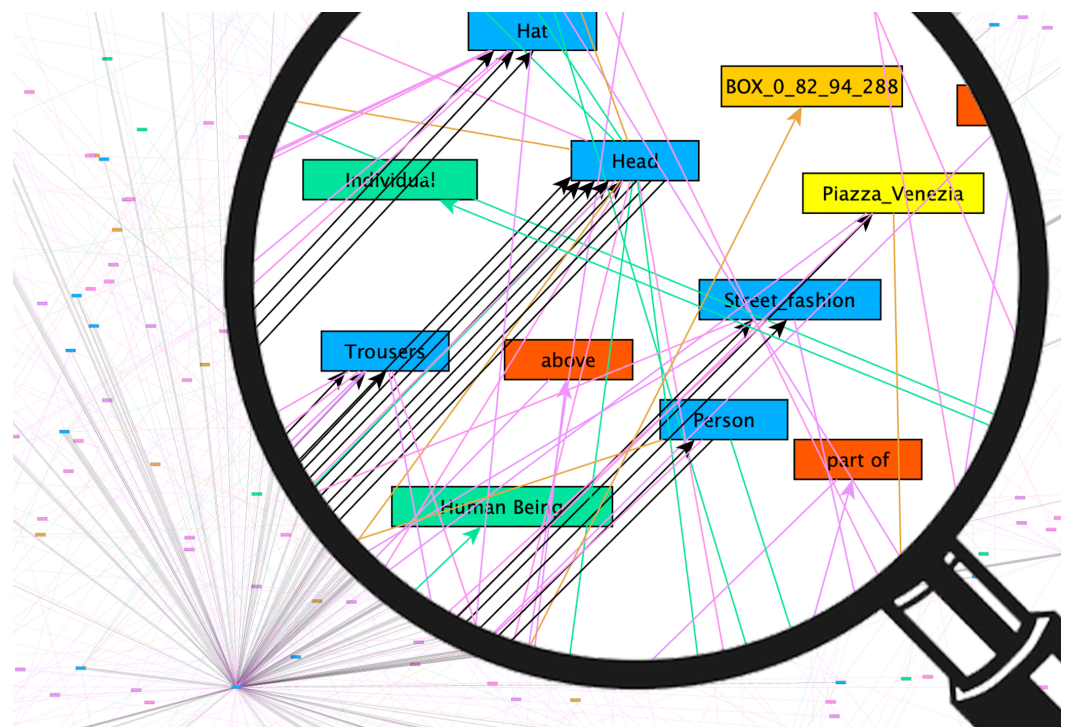


Figure 1. MMFG Visualisierung (Ausschnitt).

Das GMAF ist eine Entwicklung am Lehrgebiet für Multimedia und Internetanwendungen. Es ist in der Lage, verschiedene Multimedia-Datentypen (z.B. Bild, Text, Video) zu verarbeiten und diverse Algorithmen zur Feature-Extraktion auf die entsprechenden Dateien anzuwenden. Diese Algorithmen zur Feature-Extraktion sind erweiterbar und werden permanent durch zusätzliche GMAF-Plugins ergänzt, die dann z.B. Features in bestimmten Umfeldern extrahieren können. Eine Kerneigenschaft dieser Vorgehensweise ist, dass somit auch Features aus unterschiedlichen Quellen zu demselben Informationssatz ermittelt werden können. Ein klassisches Beispiel hierfür wäre ein Social Media Post, der neben einem hochgeladenen Bild auch eine textuelle Beschreibung und Verlinkungen zu anderen Posts enthält. Ein anderes Beispiel wäre ein Arztbrief, der sich auf MRT-Aufnahmen eines Patienten bezieht. In beiden Beispielen müssen sowohl die textuellen Informationen aus der Beschreibung und die im Bild erkannten Objekte miteinander in Verbindung gebracht werden. Im Rahmen des GMAF wird hierfür eine zentrale Datenstruktur, der sog. Multimedia Feature Graph (MMFG) verwendet, in dem Features aus unterschiedlichen Quellen und unterschiedlichen Ebenen (z.B. technische Features vs. semantische Features) verwaltet werden können. Eine Visualisierung ist in Figure 1 gegeben. Das GMAF Framework verfügt über eine Benutzeroberfläche mit entsprechenden Suchmöglichkeiten und Erklärungen, die in Abbildung 2 gezeigt ist.

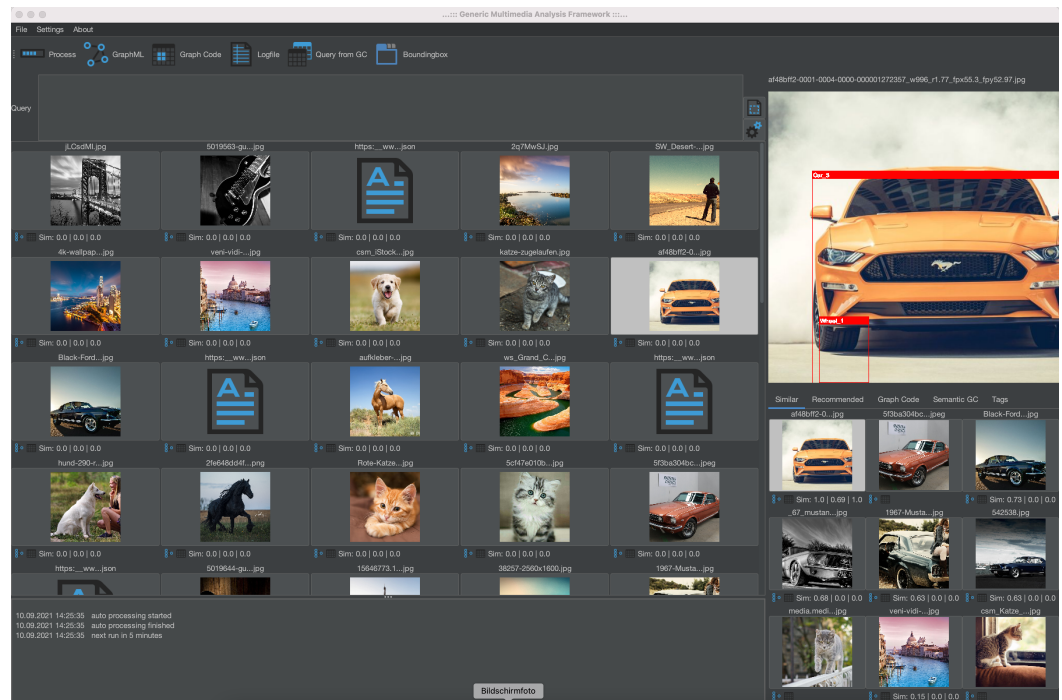


Figure 2. Benutzeroberfläche des GMAF.

Während Multimedia Inhalte im GMAF bearbeitet werden, findet im Hintergrund eine sog. Feature Fusion statt, d.h. es muss entschieden werden, welche Features relevant und welche irrelevant sind. Hierfür sind im GMAF bereits die entsprechenden Strukturen vorgesehen, so dass *FeatureFusionPlugins* entwickelt und eingebunden werden können und prozessorientierte Bearbeitungsabläufe entstehen (siehe Abbildung 3 und 4).

Eine weitere Eigenschaft des GMAF ist die *Graph Code* Technologie, ein spezielles Indexing- und Retrieval-Verfahren, das in der Lage ist, Merkmalsbasierte Anfragen schnell und effektiv zu beantworten. Im Rahmen dieser Abschlussarbeit werden Konzepte analysiert, entwickelt und implementiert, die die anwendungsfallspezifische Administration und Konfiguration des GMAF ermöglichen. Insbesondere die Komponente der Processing Flows (siehe Abbildung 4) spielt hierbei eine zentrale Rolle, da hiermit im Endeffekt die komplette Konfiguration repräsentiert werden kann. Solche Processing Flows müssen

- von einem Administrator erstellt und geändert werden können. Hier ist insbesondere die Frage nach der Editierbarkeit und der Visualisierung zu klären
- validiert und ggfs. automatisch getestet werden können
- in GMAF-taugliche Aufrufe transformiert werden
- ggfs. mit statistischen Informationen über Durchlaufzeiten oder Fehlerzustände angereichert werden

Das Ziel der Arbeit ist es, eine interaktive Benutzerschnittstelle mit "smarten" Konfigurationsmöglichkeiten zu realisieren, in der eine Konfiguration des GMAF mit mehreren Process Flows definiert und verwaltet werden kann.

2. Theoretischer Teil

Der theoretische Teil der Arbeit wird auf den Konzepten von Norman & Draper (User centered system design) [29] und Nunamaker [30] aufgebaut und methodisch hergeleitet. Als Modellierung wird die UML [31] verwendet. Dies bedeutet, dass sich auch die Dokumentstruktur von Exposé und Abschlussarbeit auf die Nunamaker Methodik stützt und entsprechende Forschungsziele der Nunamaker-Typen (Beobachtung, Theoriebildung, Implementierung und Experiment) definiert werden. Im einem Exposé formulieren Sie

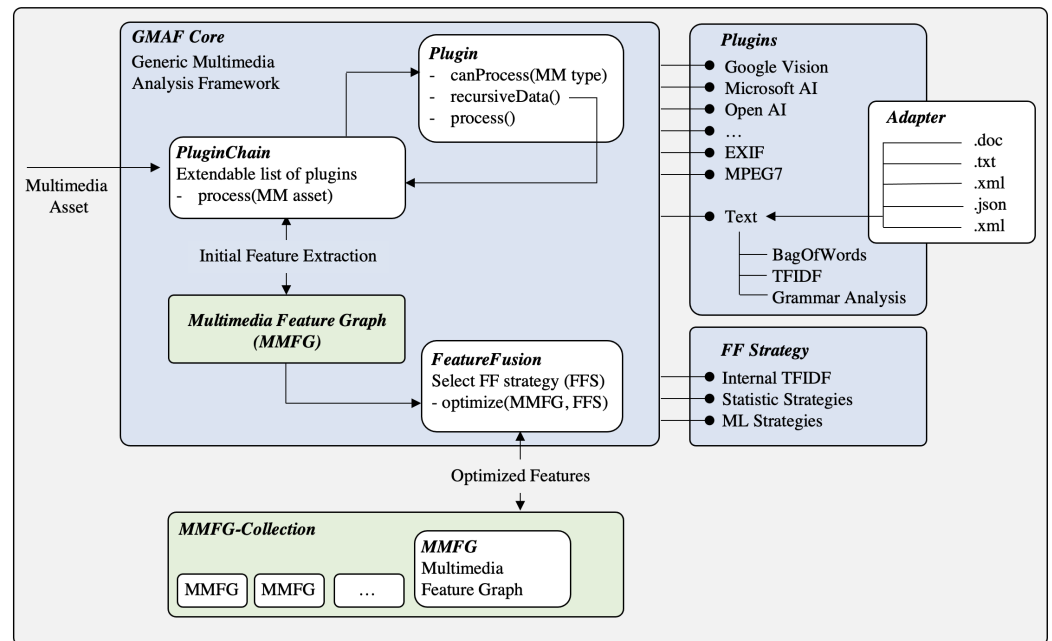


Figure 3. Feature Fusion Plugin Infrastruktur.

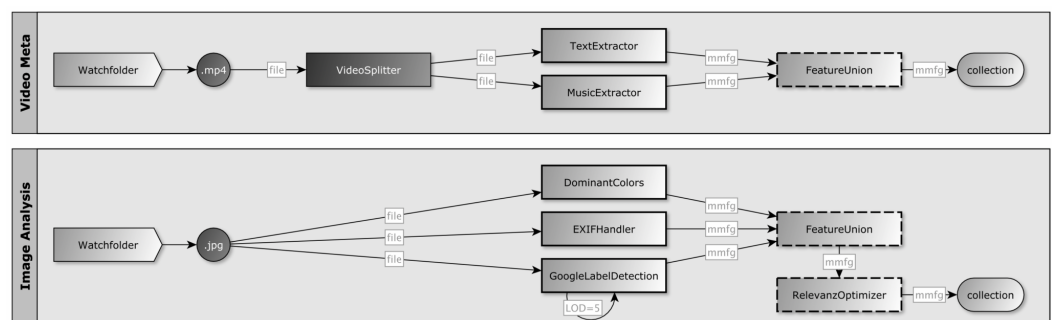


Figure 4. Processing Flows im GMAF.

zunächst Ihre Forschungsfragen und leiten die zugehörigen Forschungsziele ab. Nachdem dieses Exposé freigegeben wurde und somit auch das erste Kapitel Ihrer späteren Abschlussarbeit darstellt, führen Sie in den Stand der Wissenschaft und Technik ein und stellen die für das Thema relevanten Konzepte vor. Auf dieser Basis kann dann die Modellierung stattfinden.

3. Praktischer Teil

Der praktische Teil der Arbeit enthält die programmiertechnische Umsetzung der Modellierung und die experimentelle Überprüfung. Als Programmiersprache kommt typischerweise Java zum Einsatz. Die Verwendung von Standard-Entwicklungstools wie Eclipse, GIT, Maven, Javadoc wird vorausgesetzt. Im Rahmen des praktischen Teils erfolgt auch die Erstellung von Test-Cases und deren Dokumentation, eine Evaluierung der Ergebnisse inklusive der Wahl und Begründung einer geeigneten Evaluierungs-Methodik, sowie ein Ausblick auf zukünftige Ergänzungen. Im Rahmen der Implementierung folgen Sie den GoF Design Patterns und orientieren Ihre Arbeit an den bestehenden Strukturen. Hierbei sind insbesondere das *Command Pattern*, das *Abstract Factory Pattern*, *Singleton Strukturen* und das *Model-View-Controller Prinzip* zu berücksichtigen. Den Einsatz und die Verwendung der Patterns stimmen Sie mit Ihrem Betreuer ab.

4. Betreuung

Die inhaltliche und fachliche Betreuung der Arbeit wird von **Stefan Wagenpfeil** übernommen. Hierzu steht Ihnen jederzeit die Kommunikation per Skype, Telefon, E-Mail zur Verfügung, regelmäßige Jour-Fixes mit dem Betreuer helfen Ihnen, die Arbeitsergebnisse strukturiert voranzutreiben und geben Ihnen die Gelegenheit, durch unmittelbares Feedback zügig und effizient die Ziele der Arbeit zu erreichen. Diverse "Quality-Gates" stellen sicher, dass die Thesis auch die Anforderungen am Lehrstuhl erfüllt und somit zu einem erfolgreichen Ergebnis führen wird.

5. Aufgaben im Vorfeld

Bevor Sie sich dafür entscheiden, das Thema zu bearbeiten, sollten Sie sich einen groben Überblick zu den benötigten Voraussetzungen oder Hintergründen im Rahmen einer Recherche verschaffen. Im Literaturverzeichnis dieses Dokuments sind eine Reihe von Quellen hierzu aufgeführt. Anschließend beginnen Sie mit der Erstellung eines Exposé, welches den genauen Rahmen Ihrer Arbeit absteckt. Die Gliederung des Exposé sieht hierbei wie folgt aus:

1. Einleitung (Motivation der Arbeit)

1.1. Forschungsthema (Beschreibung des Themas, Einordnung in ein übergeordnetes Themengebiet, Relevanz der Arbeit)

1.2. Problembeschreibung (Formale Beschreibung des Problems durch die Definition von Problemfeldern)

1.3. Forschungsfragen (Definition der Forschungsfragen zu den Problemfeldern)

1.4. Methodik und Ableitung der Forschungsziele (inklusive Nunamaker-Methodik)

1.5. Ansatz (Beschreibung, wie die Forschungsziele gruppiert werden und wo sie im Rahmen der späteren Arbeit behandelt werden)

1.6. Aufbau (Festlegung der Struktur der Arbeit)

1.7. Zeitplan (inklusive Meilensteinplanung und Gantt-Diagramm auf Basis der Forschungsziele)

1.8. Zusammenfassung

Die Ergebnisse fassen Sie im Rahmen eines Dokumentes (Exposé) in ca. 10 bis 15 Seiten zusammen. Das Exposé ist die Basis für die spätere Abschlussarbeit und gibt einen für alle Beteiligten definierten Rahmen der Arbeit vor. Dadurch werden unter Anderem folgende Punkte sicher gestellt:

- Überprüfung der formalen und schriftlichen Fähigkeiten des Studenten zur Erstellung und Ausarbeitung einer Abschlussarbeit
- Vermittlung der Herangehensweise für die Strukturierung des wissenschaftlichen Diskurses
- Entwicklung eines Grundverständnisses für die zu erstellende Abschlussarbeit

Nach der Freigabe und Besprechung des Exposé mit Ihrem Betreuer und dem Lehrstuhlinhaber, beginnen Sie damit, die eigentliche Arbeit zu schreiben und die Planung, Konzeption und Implementierungen, die im Exposé bereits vorgeschlagen wurden, umzusetzen.

6. Die Abschlussarbeit

Durch die Erstellung des Exposé haben Sie bereits einen wichtigen Teil der Abschlussarbeit fertig gestellt: die saubere wissenschaftliche Verankerung, die daraus resultierende Ableitung der Forschungsziele, sowie den Ansatz und die Struktur der Arbeit. Deshalb können Sie die im Rahmen des Exposé erstellten Kapitel direkt in Ihre Abschlussarbeit übernehmen (bis einschl. 1.6). Nun erfolgt:

- eine weitergehende, selbständige Literaturrecherche
- eine Beschreibung des aktuellen Stands von Wissenschaft und Technik

- das Konzeptionelle Design der Architektur für die Umsetzung und Implementierung
- die praktische Einarbeitung (Proof-Of-Concept Implementierungen, Kennenlernen der APIs und Frameworks)
- das Erstellen von Test-Szenarios, ggfs. auch die Organisation von Test-Daten
- die Implementierung
- eine Evaluierung des Ergebnisses mit Vergleich und Ausblick

Die jeweiligen Ergebnisse halten Sie in Ihrer Abschlussarbeit fest und folgen dabei analog zu den definierten Forschungszielen einer logischen Struktur:

1. **Einleitung** (folgt dem Exposé bis einschl. 1.6)
2. **Stand der Wissenschaft und Technik** (Für jedes Forschungsziel vom Typ Beobachtung wird ein Kapitel vorgesehen und der zugehörige Stand der Wissenschaft beschrieben. Am Ende werden "remaining challenges" definiert und zusammengefasst.)
3. **Modellierung** (folgt dem Aufbau von Kapitel 2 und sieht für jedes Forschungsziel vom Typ Theoriebildung ein Kapitel vor)
4. **Implementierung** (folgt dem Aufbau von Kapitel 2 und bearbeitet Forschungsziele vom Typ Implementierung)
5. **Evaluierung** (folgt dem Aufbau von Kapitel 2 und bearbeitet Forschungsziele vom Typ Experiment)
6. **Diskussion** (Wie kann das Gesamtergebnis eingeordnet werden? Welche "remaining challenges" verbleiben? Was kann Teil von "future work" sein? Zusammenfassung)

Die Arbeit ist so ausgelegt, dass der Anteil von Theorie und Praxis jeweils 50% beträgt. Durch die hier bereits vorgeschlagene Herangehensweise werden Sie zu einer effizienten, wissenschaftlich korrekten und nachvollziehbaren Arbeitsweise angeleitet. Im Rahmen der Betreuung am Lehrgebiet steht Ihnen ein Ansprechpartner zur Verfügung, der Ihnen sowohl technisch / fachlich, als auch strukturell Hilfestellung bieten kann. Mit dem Inhaber des Lehrgebiets werden die wichtigsten Meilensteine Ihrer Arbeit ebenfalls besprochen. Sollten Sie Interesse an diesem Thema haben, kontaktieren Sie bitte einen der genannten Ansprechpartner. Vielen Dank!

References

1. Mc Kevitt et al. (2019), *Digital empathy secures Frankenstein's monster*; In CEUR Proc. of the 5th Collaborative European Research Conference (CERC 2019), Hochschule Darmstadt, University of Applied Sciences, Faculty of Computer Science, Darmstadt, Germany, March 29-30th, Vol-2348, 335-349
2. Beyerer J. (2017), *Pattern Recognition - Introduction*; Publisher: Walter de Gruyter GmbH & Co KG, Berlin; pp. 10-37, ISBN: 978-3-110-53794-9
3. Beierle C. (2019), *Methoden wissensbasierter Systeme - Grundlagen*; Publisher: Springer-Verlag, Berlin Heidelberg New York; pp. 89-157, ISBN: 978-3-658-27084-1
4. Goodfellow I. (2016), *Deep Learning*; Publisher: MIT Press, Cambridge; ISBN: 978-0-262-03561-3
5. Heaton J. (2015), *Deep Learning and Neural Networks*; pp. 1-25 ISBN: 978-1505714340
6. Google.com (2020), *Google Knowledge Search API*. Available online: <http://developers.google.com/knowledge-graph> (accessed 23.08.2020)
7. W3C.org (2020), *W3C Semantic Web Activity*. Available online: <http://w3.org/2001/sw> (accessed 23.08.2020)
8. Avola D. (2018), *Low-Level Feature Detectors and Descriptors for Smart Image and Video Analysis: A Comparative Study*; Journal: Intelligent Systems Reference Library; , pp 7-29; ISBN: 978-3-319-73890-1
9. Domingue J. (2011), *Introduction to the Semantic Web Technologies*. Available online: <https://doi.org/10.1007/978-3-540-92913-0> (accessed 2020)
10. Kwasnicka (2018), *Bridging the Semantic Gap in Image and Video Analysis*; Publisher: Springer, Berlin; pp. 97-118, ISBN: 978-3-319-73891-8
11. Mc Kevitt P. (2003), *MultiModal semantic representation*; Journal: In Proc. of the SIGSEM Working Group on the Representation of MultiModal Semantic Information, Editor Harry Bunt Kiyong Lee Laurent Romary and Emiel Kraemer
12. Spyrou (2017), *Semantic Multimedia Analysis and Processing*; Publisher: CRC Press, Boca Raton; pp.31-63, ISBN: 978-1-351-83183-3

13. Scherer R., *Computer Vision Methods for Fast Image Classification and Retrieval*; Publisher: Polish Academy of Science, Warsaw; pp. 33-134, ISBN: 978-3-030-12194-5
14. Mark Nixon, *Feature Extraction and Image Processing for Computer Vision*; Publisher: Academic Press Elsevir
15. Bhute B. (2012), *Multimedia Indexing and Retrieval Techniques: A Review*; Journal: International Journal of Computer Applications, Volume 58, pp 35-42
16. Smeulders A. (2000), *Content-based image retrieval at the end of the early years*; Journal: IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 22, pp 1349-1380
17. Gurski F. (2019), *On Characterizations for Subclasses of Directed Co-Graphs*. Available online: <http://arxiv.org/abs/1907.00801> (accessed 24.08.2020)
18. Needham M., *Graph Algorithms, Practical Examples in Apache Spark and Neo4j*; Publisher: O'Reilly; ISBN: 978-1-492-05781-9
19. Robbinson I., *Graph Databases*; Publisher: O'Reilly; ISBN: 978-1-491-93089-2
20. Jiezhong Q. (2017), *Network Embedding as Matrix Factorization: Unifying DeepWalk*. Available online: <http://arxiv.org/abs/1710.02971> (accessed 24.09.2020)
21. Bai et al. (2019), *SimGNN: A Neural Network Approach to Fast Graph Similarity Computation*. Available online: <https://doi.org/10.1145/3289600.3290967> (accessed 24.11.2020)
22. W3C.org (2013), *SPARQL Query Language for RDF*. Available online: <https://www.w3.org/TR/sparql11-overview/> (accessed 23.08.2020)
23. Nkgau T. (2017), *Graph similarity algorithm evaluation*; Journal: 2017 Computing Conference, pp 272-278
24. Sluzek A. (2013), *Local Detection and Identification of Visual Data*
25. J. S. Sevak and A. D. Kapadia and J. B. Chavda and A. Shah and M. Rahevar (2017), *Survey on semantic image segmentation techniques*; pp 306-313
26. Arndt et al. (2008), *COMM: A Core Ontology for Multimedia Annotation*; Journal of Combinatorial Theory - JCT; , pp 403-421
27. Ni J et al. (2017), *Research on Semantic Annotation Based Image Fusion Algorithm*; Journal: 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), pp 945-948
28. Gayathri N. (2019), *An Efficient Video Indexing and Retrieval Algorithm using Ensemble Classifier*; Journal: 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), pp 250-258;
29. Draper N. (1986), *User Centered System Design - New Perspectives on Human-computer Interaction*; Publisher: Taylor & Francis, Justus-Liebig-Universität; ISBN: 978-0-898-59872-8
30. Nunamaker J. (1991), *Systems Development in InformationSystems Research*; Journal of Management Information Systems 1991, pp. 89-106
31. Fowler M. (2004), *UML Distilled - A Brief Guide to the Standard Object Modeling Language*; Publisher: Addison-Wesley Professional, Boston; ISBN: 978-0-321-19368-1
32. Planche B. (2019), *Computer Vision with TensorFlow 2*; Publisher: Packt Publishing; pp. 77-99, ISBN: 978-1-78883-064-5
33. Schmitt I. (2005), *WS-QBE: A QBE-Like Query Language for Complex Multimedia Queries*; Journal: 11th International Multimedia Modelling Conference, pp 222-229
34. Jung H. (2020), *Automated conversion from natural language query to SPARQL query*. Available online: <https://doi.org/10.1007/s10844-019-00589-2> (accessed 24.10.2020)
35. Bornschlegel M. (2016), *IVIS4BigData: A Reference Model for Advanced Visual Interfaces Supporting Big Data Analysis in Virtual Research Environments*; Journal: BDA@AVI
36. Wagenpfeil S. (2020), *Github Repository of GMAF and MMFVG*. Available online: <https://github.com/stefanwagenpfeil/GMAF/> (accessed 25.09.2020)
37. Wagenpfeil S. (2021), *Towards AI-bases Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones*; SMAP 2020 Conference paper
38. Wagenpfeil S. (2021), *AI-Based Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones*; <https://www.mdpi.com/2078-2489/12/1/43>
39. Wagenpfeil S. (2021), *Fast and Effective Retrieval for Large Multimedia Collections*; <https://www.mdpi.com/2504-2289/5/3/33>
40. Wagenpfeil S. (2021), *Towards Automated Semantic Explainability of Multimedia Feature Graphs*; <https://www.mdpi.com/2078-2489/12/12/502>