

Conflict in Parliament: An Automated Analysis of the Behavior of Members of the Austrian National Council

Fabian Habersack • Adam Jatowt • Marcelo Jenny • Dominik Scherer

✉ conflict-parl-powi@uibk.ac.at



Research Goals

Contribution to political science

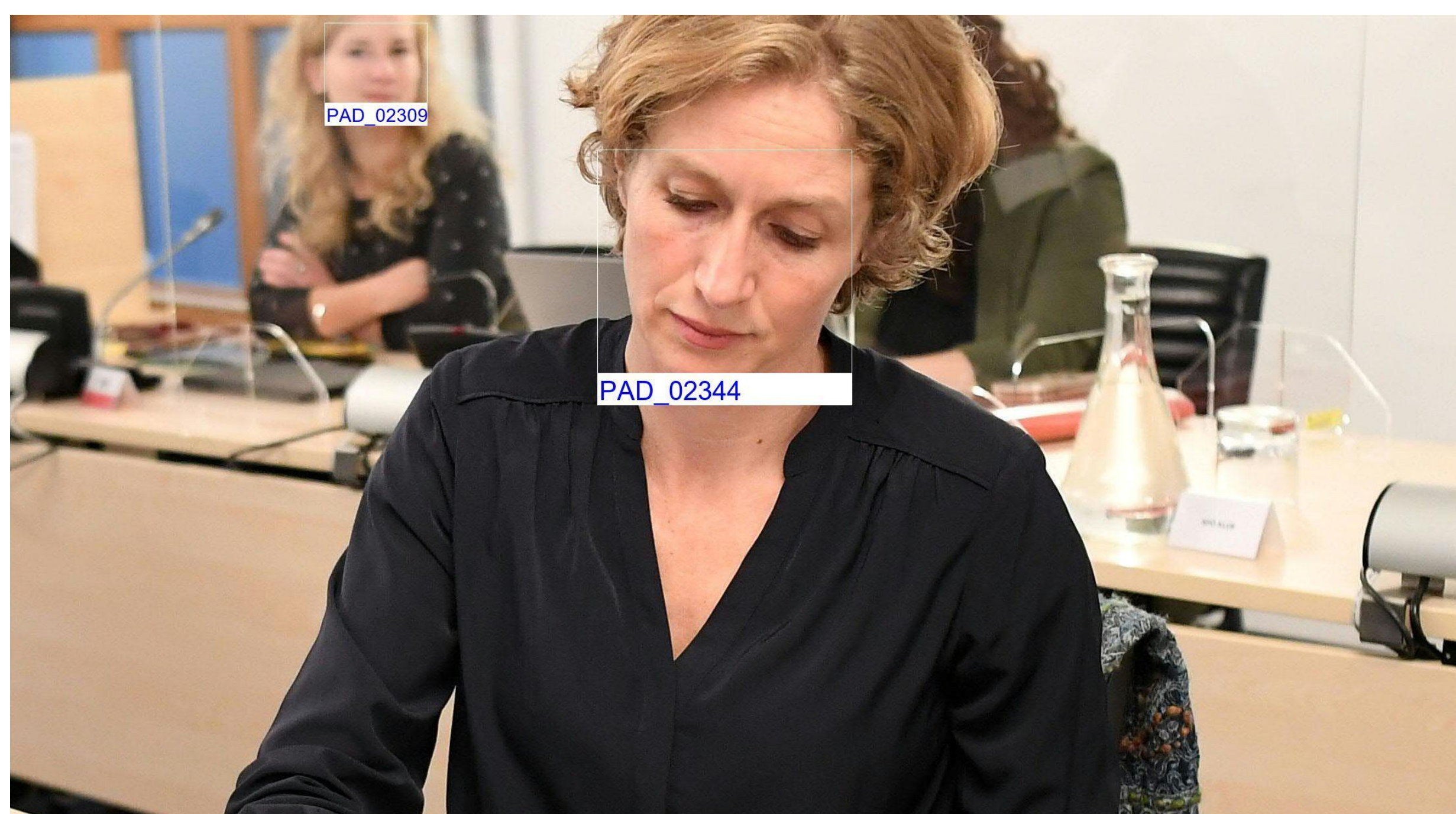
- ✓ Explaining the voting behavior of Members of the (Austrian) Parliament
- ✓ Understanding vote defections, and (strategic) absence as 'silent protest'

Methodological innovation

- ✓ Introducing *Natural Language Processing* (NLP) to the study of parliamentary behavior
- ✓ Utilizing Machine Learning and Deep Learning to record and analyze voting decisions

Societal contribution

- ✓ Archiving MP voting records via the *Austrian Social Science Data Archive*
- ✓ Providing accessible information and data for the public and researchers alike.



Workflow

1. Data Collection and Pre-Processing

- ✓ Training: photographic data from (a) Politometer; (b) Bing API; (c) parlament.gv.at
- ✓ Clean and prepare data for training of neural network algorithms

2. Face Recognition and Object detection

- ✓ Identify Members of Parliament through recognizing faces (bounding boxes)
- ✓ Identify voting decisions through object detection (sit | stand)

3. Data Analysis

- ✓ Analyze voting patterns using (automatically) recorded voting data
- ✓ Analyze other aspects of MP behavior, e.g. speeches by using emotion recognition



Date	Vote ID	MP ID	Vote (no/yes)
06-06-2021	14	PAD_00105	Yea
06-06-2021	14	PAD_93256	Yea
06-06-2021	14	PAD_01405	Nay

Data and Methods

Automated recognition of MPs in parliament using *Face recognition*

- ✓ Algorithm: *Deep Convolutional Neural Network (CNN)*; e.g. "FaceNet"; "DeepFace"; etc.
- ✓ Train pre-trained model on faces of MPs (and possibly bodies, heads, seats)

Automated identification of voting decisions using *Object detection*

- ✓ Algorithm: *Deep Convolutional Neural Network (CNN)*; e.g. "YOLOv8"
- ✓ Train pre-trained model on standing versus sitting bodies

Software to implement face detection, recognition, and vote recording

- ✓ Software: Python
- ✓ Libraries: "Face_recognition"; "DeepFace"; "Keras"; "PyTorch"; etc.



Application and Next Steps

1. Assistance with vote recording

- ✓ Determine if a bill has been passed or failed to reach required votes
- ✓ Avoid errors in aggregate and individual-level vote records

2. Analyzing voting patterns

- ✓ Investigate the effect of constituency interests on voting patterns
- ✓ Examine how far MPs ideologically deviate from party line

3. Beyond voting

- ✓ Explain emotion in MP facial expressions when speaking or voting
- ✓ Analyze audio based on audio-visual data (broadcasts of plenary debates)

